(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) EP 1 004 992 A2

(12) EUROPEAN PATENT APPLICATION

(43) Date of publication:
31.05.2000 Bulletin 2000/22

(51) Int Cl.7: G07F 7/10, G06K 19/073

(21) Application number: 00200559.3

(22) Date of filing: 24.03.1998

(84) Designated Contracting States:
AT BE CH DE DK ES FI FR GB GR IE IT LI LU MC NL PT SE
Designated Extension States:
AL LT LV MK RO SI

(30) Priority: 24.03.1997 US 41468
14.10.1997 US 61763

(62) Document number(s) of the earlier application(s) in accordance with Art. 76 EPC:
98911952.4

(71) Applicant: VISA INTERNATIONAL SERVICE ASSOCIATION
Foster City, CA 94404 (US)

(72) Inventors:
- Chan, Alfred
  Daly City, CA 94015 (US)
- Weise, Joel M.
  Burlingame, CA 94010 (US)
- Kekicheff, Marc B.
  Palo Alto, CA 94303 (US)
- Wentker, David C.
  San Francisco, CA 94117 (US)

(74) Representative: Finnie, Peter John
Elkington and Fife,
Prospect House,
8 Pembroke Road
Sevenoaks, Kent TN13 1XR (GB)

Remarks:
This application was filed on 18 - 02 - 2000 as a divisional application to the application mentioned under INID code 62.

(54) A system and method for a multi-application smart card which can facilitate a post-issuance download of an application onto the smart card

(57) A smart card includes a card domain application that manages the card. Any number of security domain applications on the card provide security for loaded applications by managing keys; each application is associated with a security domain. Each of the card domain and security domains has a command interface for off-card communication, and an API for internal card use. The card life cycle includes the states of masked, initialized, load secured and blocked. An application life cycle includes the states of not available, loaded, installed, registered, personalized, activated and blocked. An application can block the card.

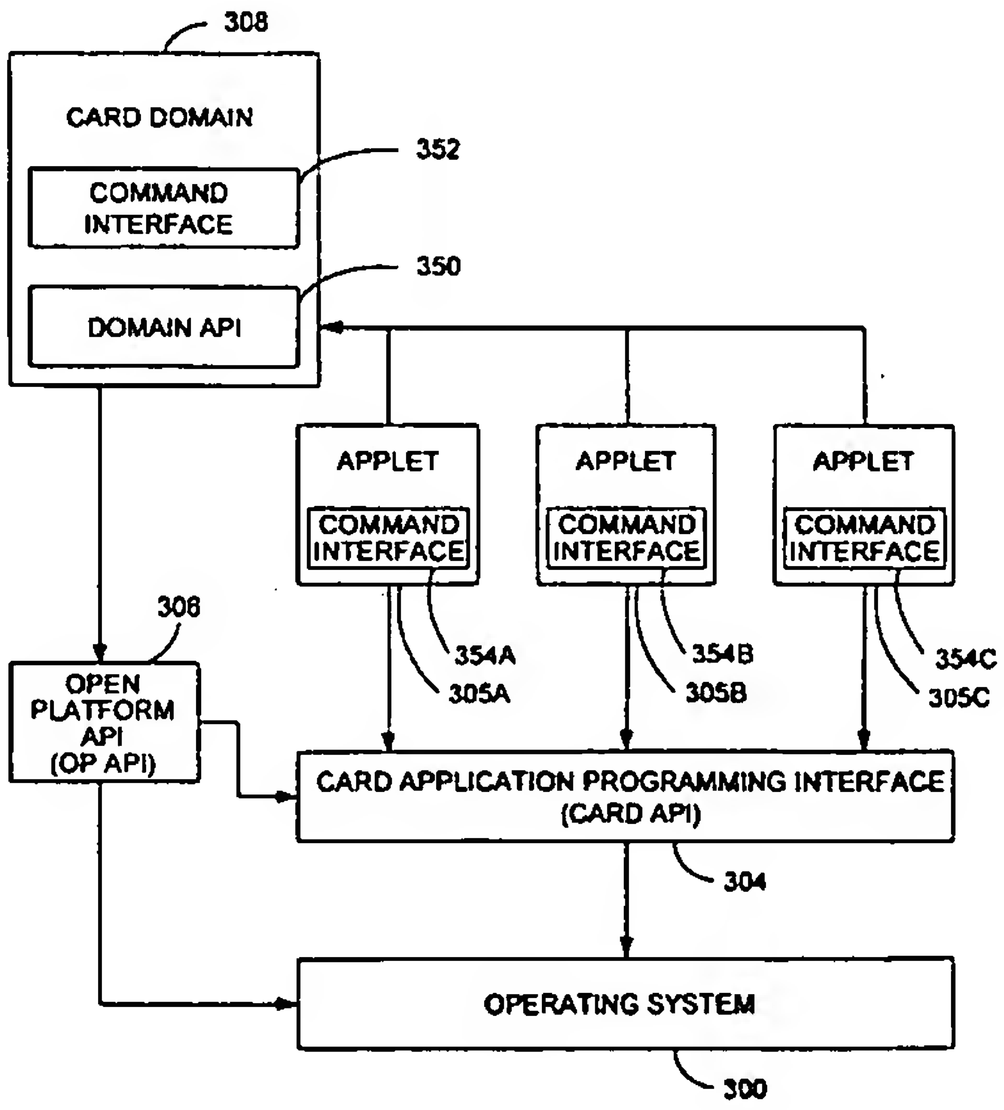


FIG. 3A

Printed by Jouve, 75001 PARIS (FR)

Description

Field of the Invention

[0001] The present invention relates to smart cards. In particular, the present invention relates to a system and method for providing a multi-application smart card which can facilitate a post-issuance download of an application onto the smart card.

Background to the Invention

[0002] A smart card is typically a credit card-sized plastic card that includes a semiconductor chip capable of holding data supporting multiple applications.

[0003] Physically, a smart card often resembles a traditional "credit" card having one or more semiconductor devices attached to a module embedded in the card, providing contacts to the outside world. The card can interface with a point-of-sale terminal, an ATM, or a card reader integrated into a telephone, a computer, a vending machine, or any other appliance.

[0004] A micro-controller semiconductor device embedded in a "processor" smart card allows the card to undertake a range of computational operations, protected storage, encryption and decision making. Such a micro-controller typically includes a microprocessor, memory, and other functional hardware elements. Various types of cards are described in "The Advanced Card Report: Smart Card Primer", Kenneth R. Ayer and Joseph F. Schuler, The Schuler Consultancy, 1993.

[0005] One example of a smart card implemented as a processor card is illustrated in FIG. 1. Of course, a smart card may be implemented in many ways, and need not necessarily include a microprocessor or other features. The smart card may be programmed with various types of functionality, including applications such as stored-value; credit/debit; loyalty programs, etc.

[0006] In some embodiments, smart card 5 has an embedded micro-controller 10 that includes a microprocessor 12, random access memory (RAM) 14, read-only memory (ROM) 16, non-volatile memory 18, a cryptographic module 22, and a card reader interface 24. Other features of the micro-controller may be present but are not shown, such as a clock, a random number generator, interrupt control, control logic, a charge pump, power connections, and interface contacts that allow the card to communicate with the outside world.

[0007] Microprocessor 12 is any suitable central processing unit for executing commands and controlling the device. RAM 14 serves as storage for calculated results and as stack memory. ROM 16 stores the operating system, fixed data, standard routines, and look up tables. Non-volatile memory 18 (such as EPROM or EEPROM) serves to store information that must not be lost when the card is disconnected from a power source but that must also be alterable to accommodate data specific to individual cards or any changes possible over the card lifetime. This information might include a card identification number, a personal identification number, authorization levels, cash balances, credit limits, etc. Cryptographic module 22 is an optional hardware module used for performing a variety of cryptographic algorithms. Card reader interface 24 includes the software and hardware necessary for communication with the outside world. A wide variety of interfaces are possible. By way of example, interface 24 may provide a contact interface, a close-coupled interface, a remote-coupled interface, or a variety of other interfaces. With a contact interface, signals from the micro-controller are routed to a number of metal contacts on the outside of the card which come in physical contact with similar contacts of a card reader device.

[0008] Various mechanical and electrical characteristics of smart card 5 and aspects of its interaction with a card reading device are defined by the following specifications, all of which are herein incorporated by reference.

[0009] Visa Integrated Circuit Card Specification, (Visa International Service Association 1996).

[0010] EMV Integrated Circuit Card Specification for Payment Systems, (Visa International Service Association 1996).

[0011] EMV Integrated Circuit Card Terminal Specification for Payment Systems, (Visa International Service Association 1996).

[0012] EMV Integrated Circuit Card Application Specification for Payment Systems, (Visa International Service Association 1996).

[0013] International Standard; Identification Cards - Integrated Circuit(s) Cards with Contacts, Parts 1-6 (International Standards Organization 1987-1995).

[0014] Prior to issuance of a smart card to a card user, the smart card is initialized such that some data is placed in the card. Initialization refers to the population of non-volatile memory with data that is common to a large number of cards while also including a minimal amount of card unique terms (e.g. card serial number and personalization keys). For example, during initialization, the smart card may be loaded with at least one application, such as credit or stored cash value, a file structure initialized with default values, and some initial cryptographic keys for transport security. Once a card is initialized, it is typically personalized. During personalization, the smart card is loaded with data which uniquely identifies the card. For example, the personalization data can include a maximum value of the card, a personal identification number (PIN), the currency in which the card is valid, the expiration date of the card, and cryptographic keys for the card.

[0015] A limitation of conventional smart cards is that new applications typically can not be added to an issued smart card. Smart cards are traditionally issued with one or more applications predefined and installed during the manufacturing process of the card. As a result, with traditional smart card implementation, once a card has been issued to a card user, the smart card becomes a fixed application card. If a new application is desired, the smart card is typically discarded and a new smart card, which includes the new application, is issued.

[0016] It would be desirable to provide a smart card which would allow applications to be loaded after the card is issued. Further, it is desirable to provide a mechanism to manage the loading of an application as well as general management of the applications on the smart card. Additionally, it is desirable to allow an application provider to keep cryptographic keys confidential from the issuer of the smart card and to securely allow applications from different entities to coexist on a card.

Summary of the Invention

[0017] The present invention provides a system and method which allow card issuers to add applications during the lifetime of the card after the card has already been issued (referred to herein as post issuance loading). Downloading an application after the card has been issued to the card holder will be referred to herein as a "secure install" process.

[0018] The system and method of the present invention allow the post issuance loading of an application and/or objects from an application server via a card acceptance device and its supporting system infrastructure delivery mechanism onto a card in a secure and confidential manner.

[0019] The present invention also provides a system and method for controlling at least one function associated with an issued smart card. In a multi-application smart card, a privileged application, herein referred to as a card domain, manages multiple functions related to the smart card. Examples of these functions include card initialization, global card data, card life cycle, and secure installation of smart card applications.

Brief Description of the Drawings

[0020] Examples of the present invention will now be described in detail with reference to the accompanying drawings, in which:

Figure 1 is a block diagram of a smart card system suitable for implementing the present invention;
Figure 2 is an example of a block diagram of software layers which can be utilized in a smart card;
Figures 3A - 3B are block diagrams of examples of software layers according to embodiments of the present invention;
Figure 4 is a flow diagram of an example of a method according to an embodiment of the present invention for installing an application onto an issued smart card utilizing a card domain;
Figure 5 is a flow diagram of a method according to an embodiment of the present invention for providing confidential information to an application in a smart card using security domains;
Figure 6 is a flow diagram of an example of a method according to an embodiment of the present invention for installing an application onto an issued smart card utilizing a card domain;
Figure 7A is a flow diagram illustrating a sequence of card life states;
Figure 7B is a flow diagram illustrating a sequence of card life states;
Figure 8 is an illustration of an example of a card life cycle;
Figure 9 is a flow diagram of an example of a method according to an embodiment of the present invention for blocking a card utilizing a card domain;
Figure 10 is a block diagram illustrating interactions between a card domain and a security domain on a smart card according to an embodiment of the present invention;
Figures 11A and 11B are flow diagrams of an example of a method according to an embodiment of the present invention for loading an application by using a security domain after the smart card has issued;
Figures 12A-12B are flow diagrams of an example of a method according to an alternate embodiment of the present invention for loading an application using a security domain after the smart card has issued; and,
Figure 13 is a block diagram illustrating an example of key management and key dependencies for post issuance download of applications onto the smart card.

Detailed Description

[0021] Figure 2 is a block diagram of an example of software layers which can be utilized in a smart card. The smart card shown in Figure 2 includes an operating system 200, a card application programming interface (API) 204, and applications 206A-206B. Operating system 200 can include functionality to control the cards, memory management, input/output (I/O), and cryptographic features. Card API 204 utilizes the instructions from operating system 200 and writes these instructions into blocks which can be reused for common routines in multiple applications. Applications 206A and 206B can run on the smart card via instructions from API 204. These applications can include any application which can run on a smart card, such as stored value, credit, debit, transit, and loyalty.

[0022] One embodiment of the present invention is based upon the Java Card standard. In this case applications are referred to as 'Applets' and they are written to link to a Java Card API which is the application programming interface present on smart cards built to the Java Card standard.

[0023] Although the conventional software system shown in Figure 2 allows for multiple applications, it does not solve the problem of how to securely load an application after issuance of the smart card to a user. If an application is to be loaded post issuance, a mechanism is needed to manage the loading of an application as well as the general management of the applications on the smart card. Additionally, an application provider may wish to keep cryptographic keys confidential from the issuer of the smart card. Accordingly, a mechanism is needed to provide for the separation of confidential information between an application provider and an issuer of a smart card. Embodiments of the present invention address such a need.

[0024] Figures 3A - 3B are block diagrams showing software components of a smart card according to embodiments of the present invention. The arrows indicate dependencies between components. Figure 3A shows an embodiment of a smart card utilizing a card domain, while Figure 3B shows an embodiment of a smart card utilizing a security domain, as well as a card domain.

[0025] The example shown in Figure 3A includes an operating system 300, a card API 304, applications 305A-305C, a card domain 308, and open platform (OP) API 306. The system shown in Figure 3 allows for a secure and managed post issuance download of an application onto a smart card. A card domain is a card issuer's on-card control mechanism for a smart card according to the present invention.

[0026] Open platform API 306 classifies instructions into card domain 308 and security domains 310A-310B (shown in Figure 3B). Accordingly, OP API 306 facilitates the formation of instructions into sets which can be identified as being included as part of card domain 308 and security domains 310A-310B.

[0027] Applications 305A-305C can include any application which can be supported by a smart card. Examples of these applications include credit, debit, stored value, transit, and loyalty. Applications 305A-305C are shown to include command interfaces, such as APDU interfaces 354A-354C which facilitate communication with the external environment.

[0028] Applications 305A-305C can run on the smart card via instructions from card API 304. Card API 304 is implemented using the instructions from the card operating system and writes these instructions into blocks which can be reused for common routines for multiple applications. Those skilled in the art will recognize that a translation layer or interpreter may reside between API 304 and operating system 300. An interpreter interprets the diverse hardware chip instructions from vendor specific operating system 300 into a form which can be readily utilized by card API 304.

[0029] Card domain 308 can be a "privileged" application which represents the interests of the smart card issuer. As a "privileged" application, card domain 308 may be configured to perform multiple functions to manage various aspects of the smart card. For instance, card domain 308 can perform functions such as installing an application on the smart card, installing security domains 310A-310B (shown on Figure 3B), personalization and reading of card global data, managing card life cycle states (including card blocking), performing auditing of a blocked card, maintaining a mapping of card applications 305A-305C to security domains 310A-310B, and performing security domain functions for applications 305A-305C which are not associated with a security domain 310.

[0030] Card domain 308 is shown to include an API 350 and a command interface, such as Application Protocol Data Unit (APDU) interface 352. APDU interface 352 facilitates interfacing with the external environment in compliance with, e.g., International Standards Organization (ISO) Standard 7816-4, entitled "Identification Cards - Integrated circuit(s) cards with contacts - Part 4, Inter-industry commands for interchange," which is herein incorporated by reference.

[0031] For example, APDU interface 352 can be used during post issuance installation of an application or during loading of card global data. An application load and install option is performed via a set of appropriate APDU commands received by card domain 308. API 350 facilitates interfacing with the internal smart card environment. For example, API 350 can be used if card domain 308 is being utilized as a default in place of a security domain 310, or if an application requires information such as card global data, key derivation data, or information regarding card life cycle.

[0032] Memory allocations have been performed by the time an application is in an install state. An application is also personalized after loading and installing. A personalized application includes card holder specific data and other required data which allows the application to run. In addition to managing the installation and personalization of the application, card domain 308 can also manage global card information. Global card information includes information that several applications may need to perform their functions, such as card holder name and card unique data utilized in cryptographic key derivations. Card domain 308 can be a repository for the global card information to avoid storing the same data multiple times.

[0033] Card domain 308 can also manage card life cycle states including card blocking. The smart card will typically move through several states during its life cycle. Card domain 308 keeps track of what state the card is in during its life cycle. Card domain 308 may also manage a block request to block virtually all functions of the card. Further details of card domain 308 management of a block request will be discussed in conjunction with Figure 6. Card domain 308 may also keep track of the state of an application during an application's life cycle. This kind of information regarding an application can be utilized during an auditing of a card. Auditing can be performed at any time during a card's lifetime. For instance, auditing may be performed after a card has been blocked or prior to installing a new application to validate the card contents. Although virtually all card functions are no longer functioning when a card is blocked, an issuer may be able to query card domain 308 for information regarding a state of an application or the life cycle state of the card. In this manner, the issuer of a card may still access a profile of the blocked card and its applications.

[0034] Figure 3B shows an embodiment of the present invention utilizing a security domain 310, as well as card domain 308'. The example shown in Figure 3B includes an operating system 300', a card API 304', applications 305A'-305C', security domains 310A-310B, a card domain 308', and open platform (OP) API 306'. The system shown in Figure 3B also allows for a secure and managed post issuance download of an application onto a smart card.

[0035] Card domain 308' can work in conjunction with a security domain 310. Security domain 310 is a logical construct that can be implemented as an application to provide security related functions to card domain 308' and to applications associated with security domain 310. Security domains 310A-310B can assist in secure post issuance loading of an application onto the smart card. Security domains 310A-310B provide for a mechanism which keeps the application provider's confidential information, such as cryptographic keys, from being disclosed to the issuer of the smart card.

[0036] There may be multiple security domains 310 on a smart card, each represented by a unique cryptographic relationship. A security domain 310 is responsible for the management and sharing of cryptographic keys and the associated cryptographic methods which make up the security domain's cryptographic relationship. An application which is loaded to the smart card post issuance can be associated with a security domain, preferably with only one security domain. However, multiple applications may be associated with the same security domain 310. Applications installed on a smart card during the pre-issuance phase may optionally be associated with a security domain 310 on the smart card for purposes of loading confidential personalization data to those applications using security domain 310 keys.

[0037] The software for security domain 310 may be installed by the card manufacturer at the time of card manufacturing (e.g., when the ROM is masked), or may be added during initialization or personalization stages. Security domains 310 can be implemented as selectable applications which are isolated from one another and the rest of the system. If security domain 310 is implemented in a Java card as an application, standard Java card security can be relied upon to ensure isolation of security domain 310. In addition, or alternatively, other security mechanisms such as hardware security can be utilized through OP API 306 implementation. OP API 306 may utilize special security features to enforce isolation of security domain 310. An example of such a security feature is the utilization of chip hardware security routines which may be employed by OP API 306.

[0038] Each security domain 310A-310B provides a command interface, such as an Application Protocol Data Unit (APDU) interface 320A-320B, for communication off card, and on card APIs 322A-322B.

[0039] The APDU interface 320A or 320B consists of personalization commands and is intended to allow the initial loading of security domain keys and to support key rotation if desired during the life of the security domain. APIs 322A-322B may include a signature verification method and decryption method which are shared with card domain 308' for post issuance loading of applications. Additionally, applications may utilize API interfaces 322A-322B for decrypting application confidential data. Note that card domain 308' may always function as a security domain and does so as the default.

[0040] Security domain 310 manages signing and decrypting keys and provides cryptographic services using those keys. Security domain 310 processes APDU's for numerous functions. These functions can include key management functions, e.g., functions to load or update keys. During a secure installation of an application, security domain 310 can provide services to card domain 308' to decrypt an application install file and check the signature of an application file. For an application associated with a security domain 310, that application's security domain 310 provides decrypt and signature functions, such as MACing on an update key APDU command during the personalization phase of a newly installed application. Thereafter, the application can use the updated key to decrypt and check signatures on subsequent key updates.

[0041] The smart card issuer may decide whether security domain 310 utilizes a static key or a session key for transactions. A static key is a cryptographic key which exists prior to processing APDUs and which exists during and after the processing of APDUs. A session key is a cryptographic key which can be generated for a particular transaction and is typically no longer used for APDU processing after the transaction. If a session key is utilized, security domain 310 preferably derives its own session key for processing APDUs.

[0042] Figure 4 is a flow diagram of a method according to an embodiment of the present invention for providing an application to a smart card. The example illustrated in Figure 4 also applies to installing a security domain 310 onto a smart card. Note that all of the flow diagrams in this application are merely examples. Accordingly, the illustrated steps of this and any other flow diagram , can occur in various orders and in varying manners in order to accomplish virtually the same goal.

[0043] A smart card is issued (step 400), and an application is forwarded to the issued smart card (step 402). The forwarding of the application can occur through any electronic media which can interface with a smart card and connect to an appropriate network. For example, devices such as an automatic teller machine (ATM), a display phone, or a home computer, can be used to forward an application to the issued smart card. The forwarded application is then loaded onto the smart card, and the loading of the application is managed by card domain 308 (step 404).

[0044] Figure 5 is another flow diagram of a method according to an embodiment of the present invention for providing an application onto an issued smart card. A smart card is created and provided with a first application, the first application including a cryptographic service (step 1002). A second application is loaded onto the smart card (step 1004). Thereafter, the second application is installed, and the cryptographic service of the first application is utilized to install the second application (step 1006).

[0045] Figure 6 is another flow diagram of an example of a method according to an embodiment of the present invention for providing an application onto an issued smart card. This method for providing an application also applies to providing a security domain 310 onto the smart card. In the example shown in Figure 6, a card issuer deploys smart cards to customers (step 500). A decision is made to install a vendor's application onto the issued smart card (step 502). When a dialogue between the issuer and the smart card is initiated, a presigned copy of the application is forwarded to the smart card (step 504). As previously stated, the dialogue between the issuer and the smart card can occur via any electronic device which can interface with a smart card and connect to an appropriate network. The application can be pre-signed with a key equivalent to that which already exists on the card so that each application has a unique signature that can be verified by the card.

[0046] Card domain 308 can then take the steps to load the application. Card domain 308 decrypts the forwarded application and checks the signature of the application (step 508). Card domain 308 can decrypt the application with the issuer's secret key. An appropriate cryptography method, such as Data Encryption Standard (DES) or 3DES, can be utilized to decrypt at least a portion of the application. Those skilled in the art will recognize that a number of cryptographic techniques may be used to implement embodiments of the present invention. For the purpose of illustration, symmetric key techniques are addressed herein, although asymmetric techniques are also contemplated. A good general cryptography reference is Schneier, Applied Cryptography, 2d Ed. (John Wiley, 1996), the contents of which are incorporated herein by reference.

[0047] It is then determined whether the signature on the application is valid (step 510). If the signature associated with the application is not valid, then the application is not loaded onto the card and the process ends (step 520). If, however, the signature associated with the application is valid the application is then installed and available for personalization. During personalization the application receives personalization data (step 512). Personalization data includes data which is unique to the smart card user. For instance, in a airline loyalty application, personalization data can include the smart card user's seating preference, meal preference, and eligibility for various possible perquisites. This personalization data can also be signed and encrypted.

[0048] The application then invokes card domain's 308 decryption service for the received personalization data (step 513). Card domain 308 can then performs a signature check for the received personalization data (step 514). Methods of decrypting personalization data and performing signature checks are well known in the art. Finally, the application can then be activated (step 518).

[0049] A new application which as been downloaded onto a smart card post-issuance can be stored in a variety of ways. One example is to store the application into a file. Another example is to maintain a pointer to the application object.

[0050] Figure 7A is a flow diagram illustrating an example of card life cycle. The sequence is preferably considered irreversible. The first card state is when the smart card is Masked (700). During the Masked state (700), the smart card obtains its operating system, card identification, and preferably at least one application. The Masked state (700) is achieved as soon as all of the necessary components for card initialization are made available. An example of when necessary components are made available is when card domain 308 and OP API 306 are enabled, as well as the Java card environment being enabled, such as a Java card virtual machine and a Java card API.

[0051] After the Masked state, the next state is the Initialized (702) state. The Initialized state is achieved once all card activity requiring an initialization key is complete. As part of card initialization, if not already available, the card domain 308 application must be installed and registered. In addition , one or more security domains may also be installed and registered. These installed domains must then be selected and personalized. An initialization key is a secret key which is typically used by a smart card manufacturer during loading of data onto the smart card prior to issuance.

[0052] The next state is Load Secured (704). The Load Secured state is achieved after a secure install (post-issuance download) mechanism for loading of applications through the remainder of the card lifetime has been established.

[0053] The final card state is when the card is either expired or blocked (706). The blocked state is achieved as soon as an authorized smart card application has received a command to block the card.

[0054] The card life cycle is preferably an irreversible sequence of states with increasing security. Initialized and all subsequent card life cycle states and their transitions are preferably under the control of card domain 308. Card domain 308 executes and responds to commands that result in a transition in the card life cycle from one state to the next. These commands are preferably Application Protocol Data Unit (APDU) commands. Card domain 308 is also responsible for the installation of applications on the card, but preferably has no control over the applications' life cycle states. Each application is preferably responsible for its own application life cycle state management but it preferably allows card domain 308 to have access to its life cycle states for auditing purposes.

[0055] The card life cycle is designed in such a way to increase the level of security enforced by the card at each successive state. As stated above, the cycle is also established as a process which can only ratchet forward to ensure that once the card begins a life cycle state with associated security policies, the only option is to cycle forward to the next state in the life cycle with a higher level of security. The card domain as the system security manager of the card maintains the current life cycle state, enforces the associated security policies, and controls the state transitions in the card life cycle.

[0056] Figure 7B is a flow diagram illustrating an example of an application life cycle. The application is initially unavailable (750). This state is achieved as soon as an applet is deleted and/or blocked from access. The next state is a loaded state (752). The application reaches the loaded state once the application has been physically loaded onto the smart card. The application is then installed (754), and registered (756). The installed state is achieved as soon as an applet has allocated all of the necessary space and data structures for operation. The registered state is achieved as soon as an applet has been included in the registry of applets. Once the application is registered, it can be deleted at any time thereafter. The next state is the personalized state, wherein personalized information is included in the application (758). The personalized state is achieved as soon as an applet has been updated with card holder specific data. Finally, the application may expire or be blocked (760).

[0057] Figure 8 is an illustration of an example of multi-application card life time line. The illustration does not include any off-card activities that occur prior to masking, initialization, personalization, Secure Install, nor activities that may be related to card management procedures. The illustration shows a card that contains six applications (A through F) over its lifetime.

[0058] In this example, application A can be installed in ROM and used during the complete life of the card from Masked ROM stage 800 to card blocked/expired stage 802. Application B is also in ROM and utilized during a first portion of the life of the smart card. The life of application B ends at stage 804. Application C is located in non-volatile memory, such as EEPROM, which is loaded during initialization. Application C is shown to expire at stage 806. Application D is also located in EEPROM and is used for the complete life of the card until card blocked/expired stage 802. Application E is installed at stage 808, sometime after issuance of the smart card. Application E is located in EEPROM and used until the end of the card life at card blocked/expired stage 802. Application F is also installed post issuance at stage 810, and expires sometime before the end of the card life at stage 812.

[0059] Figure 9 is a flow diagram of a method according to an embodiment of the present invention for blocking a card. A card be can be blocked if a breach of security is detected by an application. According to an embodiment of the present invention, a smart card can be blocked while an application is in use. A blocked card will no longer operate so that a suspect user cannot utilize any of the applications on the smart card. Blocking is merely one example of the many functions card domain 308 can perform in managing the other applications on the smart card. Examples of other functions include installing an application on the smart card, installing security domains 310A-310B, personalization and reading of card global data, managing card life cycle states including card blocking, performing auditing of a blocked card, maintaining a mapping of card applications to security domains, and performing security domain functions for applications which are not associated with a security domain.

[0060] In the example shown in Figure 9, an application is currently in use (step 600). The application detects a problem which triggers a card block request from the application (step 602). The application then sends a card block request to card domain 308 (step 604). Card domain 308 determines whether the card block request is valid (step 606). A card block request can be valid if the request originates from a predetermined application. If the card block request is not valid, the card domain 308 does not block the smart card (step 608). However, if the card block request is valid, then card domain 308 authorizes the card blocking (step 610), and card domain 308 blocks the smart card (step 612) such that the smart card will reject any attempted transactions for any of the applications on the card.

[0061] Figure 10 is a block diagram illustrating the use of security domain 310 by the card domain 308. The method and system according to an embodiment of the present invention allows for multiple application providers to be represented on a smart card in a secure and confidential manner. This security and confidentiality can be achieved through the use of security domains 310A-310B shown in Figure 3.

[0062] Figure 10 illustrates an example of a smart card which contains two security domains 310A-310B. In this example, it is assumed that a masked application 305A from the smart card is associated with a security domain, such as security domain 310A, and an additional application 305B will be added post issuance and be associated with a second security domain, such as security domain 310B. The arrows indicate key relationships between the various smart card entities as will now be described. Masked application 305A uses key services from security domain 310A for decrypting confidential data and optionally for full personalization. Card domain 308 uses key services from security domain 310B for decrypting and checking the signature of an application loaded post issuance, such as post issuance loaded application 305B. Post issuance loaded application 305B uses key services from security domain 310B for decrypting confidential data and optionally for full personalization.

[0063] Figures 11A and 11B are further flow diagrams of an example for a method according to an embodiment of the present invention for providing an application onto an issued smart card. The card issuer decides to include a security domain 310 onto a smart card (step 1100). The issuer assigns security domain 310 to vendor A (step 1102). Vendor A, or an application developer on behalf of vendor A, generates cryptographic keys such as those used in symmetric or asymmetric cryptography operations (step 1104). Examples of these cryptography operations include encryption, decryption, MACing, hashing, and digital signatures. Examples of cryptographic methods which utilize such keys and are suitable for implementation for the embodiment of the method and system of the present invention include Data Encryption Standard (DES) and 3DES. The card personalization agent receives the keys and loads security domain keys associated with a specific security domain 310 for each smart card. (1106). The card personalization agent receives smart cards and collects other data, OS, code, and application and card holder specific data, and places the data on the smart card (step 1108).

[0064] The card issuer then deploys the smart card to customers (step 1110). A decision is then made to install vendor A's application on the smart card (step 1112). When a dialogue between the smart card issuer and the smart card is initiated, a signed copy of the application is forwarded to the smart card (step 1114). The application can be signed with a key equivalent to that which already exists on the smart card so that each application has a unique signature that can be verified by the smart card.

[0065] The smart card's card domain 308 then takes steps to load the application. Card domain 308 invokes an associated security domain's cryptographic service to decrypt the application and check the signature (step 1118). It is then determined if the signature is valid (step 1120). If the signature is not valid, the process ends (step 1122). If, however, the signature is found to be valid, then the application receives personalization data which can be signed and optionally encrypted (step 1124). The loaded application then invokes its associated security domain's decryption service and signature check for the received personalization data (step 1126). Secret keys required to run or operate the application on the smart card are used to activate the application by authentication (step 1130).

[0066] Figures 12A and 12B are flow diagrams of a method according to another embodiment of the present invention for providing confidential information to an application using a security domain 310. The issuer decides to include a security domain 310 on a smart card (step 1200). A trusted party generates secret cryptographic keys and sends the keys to a card personalization agent in a secure manner (step 1201). A trusted party is typically a third party who performs the function of certifying the source of information, such as a signature. A card personalization agent (which may be the same as the trusted party) receives the key and loads a unique secure domain key associated with a specific security domain 310 for each smart card (step 1202).

[0067] The card personalization agent receives the smart card and collects other data, such as OS, code, and application and card holder specific data, and places the data on the smart card (step 1204). The issuer then deploys the smart card to its customers (step 1206). A decision is made to install vendor A's application on the issued smart card (step 1208). Vendor A obtains secret keys for security domain 310 from the trusted party (step 1210). Vendor A then sends the smart card issuer a signed copy of Vendor A's application (step 1212).

[0068] When a dialogue between the smart card issuer and the smart card is initiated, a signed copy of the application is forwarded to the smart card (step 1214). The application can be signed with a key equivalent to that which already exists on the smart card so that each application has a unique signature that can be verified by the smart card. Card domain 308 invokes security domain's cryptographic service to decrypt the associated application and check its signature (step 1218). It is then determined whether the signature is valid (step 1220). If the signature is not valid, then the process ends (step 1222).

[0069] If, however, the signature is valid, then the application receives personalization data, which can be signed and optionally encrypted (step 1224). The loaded application then invokes security domain's decryption service and signature check for the received personalization data (step 1226). The cryptographic secret data required to run or operate the application on the card are used to activate the application (step 1230).

[0070] Figure 13 is a block diagram illustrating the use of cryptographic keys for post issuance loading of an application onto a smart card. Applications that are not masked and not loaded during card initialization stage or personalization stage need their executables downloaded using a secure installation method, such as the post issuance download described in the previous figures. The applications can be loaded using the card domain cryptographic keys. The applications are then decrypted and can have their signature verified using the key services of the corresponding security domain 310. Therefore, the desired security domain(s) 310 preferably have encryption and signature keys installed prior to the post issuance download of the corresponding application.

[0071] In the example shown in Figure 13, only one security domain 310 is shown since security domains 310 for other applications are not relevant to illustrate the downloading of a single application. Note that the result of the secure installation is initially a loaded application, which must then be installed, registered and personalized. After loading, the application is installed, preferably by issuing an install APDU command to card domain 308. An application can be installed when its install method has executed successfully. Memory allocations have been performed by the time an application is in an install state. A loaded application should also be registered. When an application is registered, it is selectable and it is ready to process and respond to APDU commands. Installation and registration may be performed simultaneously by the same APDU command. An application is also personalized after loading. A personalized application includes cardholder specific data and other required data which allows the application to run.

[0072] In the example shown in Figure 13, the cryptographic key and MAC/Signature key are shown to be included in the functions of card domain 308/security domain 310. If a security domain is associated with the application being loaded, then the security domain will be invoked. However, if no security domain 310 is associated with the application which is being loaded, then the cryptographic key and the signature key of card domain 308 will be utilized. In contrast to the install commands sent to the smart card during the initialization phase, the post issuance install command is not issued in a secured environment, therefore it is preferably protected with a cryptographic key, such as a MAC/Signature key. Card domain 308 manages the post-issuance loading of a new application, while security domain 310 ensures the validity and integrity of the new application once the new application has been loaded onto the smart card. If a security domain 310 is not associated with the newly loaded application, then card domain 308 performs security domain's 310 functions. Once the new application is post-issuance downloaded, various keys such as a cryptographic key and a signature key are preferably utilized for installation and personalization of the application.

[0073] A method and system for a smart card domain and a security domain has been disclosed. Software written according to the present invention may be stored in some form of computer-readable medium, such as memory or CD-ROM, or transmitted over a network, and executed by a processor.

Claims

1. A smart card comprising:

   a card life cycle having a plurality of states;
   a memory including an indication of which of said states said card life cycle is in; and
   a card domain application including

      an issuer key associated with the issuer of said smart card,
      a function for managing said life cycle of said smart card, and
      a function for tracking the status of said life cycle of said smart card, whereby said card domain application represents the interests of the issuer and manages said card life cycle.

2. A smart card according to claim 1, wherein said card domain application further includes:
   a function for blocking said smart card.

3. A smart card according to claim 1 or 2, wherein said states of said card life cycle include masked, initialized, load secured and blocked.

4. A smart card according to any preceding claim, wherein said states of said card life cycle are in an irreversible sequence.

5. A smart card according to any preceding claim, wherein the contents of said memory determines the state of said card life cycle.

6. A method of blocking a smart card comprising the steps of:

   detecting a problem with said smart card by an application of said smart card; sending a card block request from said application to a card domain application of said smart card, said card domain application having the capability to block said smart card;
   determining by said card domain application whether said card block request is valid; and,
   blocking said smart card by said card domain application, whereby said smart card is not operational for a user.

7. A method according to claim 6, wherein said card domain application includes an issuer key associated with the issuer of said smart card, whereby said card domain application represents the interests of the issuer.

8. A method of moving a smart card through a sequence of card life cycle states, said method comprising the steps of:

   receiving said smart card in a masked state, said masked state indicating that components necessary for initialization are available on said smart card;
   initializing said smart card using an initialization key;
   placing said smart card into an initialized state;
   loading an application onto said smart card post-issuance; and,
   placing said smart card into a load secured state, whereby said smart card passes through

a number of said states of said card life cycle.

9. A method according to claim 8, further comprising the steps of:

receiving a card block request;
blocking said smart card; and,
placing said smart card into a blocked states, whereby said smart card is not operational for a user.

10. A method according to claim 8 or 9, wherein said states of said card life cycle are in an irreversible sequence.

11. A method according to any of claims 8 to 10, wherein said states of said card life cycle are in an irreversible sequence.

12. A method according to any of claims 8 to 11, wherein said states of said card life cycle place said smart card into an increasing level of security.

13. A smart card comprising:

a first application having a sequence of life cycle states; and
a card domain application including

an issuer key associated with the issuer of said smart card,
a function for loading said application onto said smart card, said installing causing said first application to be placed into an installed state; and,
a function for registering said application on said smart card, said registering causing said first application to be placed into a registered state, whereby said card domain application represents the interests of the issuer and manages said first application.

14. A smart card according to claim 13, wherein said card domain application further includes:
a cryptographic service for loading said first application onto said smart card post-issuance.

15. A smart card according to claim 13 or 14, wherein said first application further includes:
a function for personalizing said first application, said personalizing causing said first application to be placed into a personalized state, whereby said personalizing is under the authority of said first application.

16. A smart card according to any of claims 13 to 15, wherein said first application further includes:

a function for blocking said first application said blocking causing said first application to be placed into a blocked state, whereby said blocking is under the authority of said first application.

17. A method of moving an application of smart card through a sequence of application life cycle states, said method comprising the steps of:

receiving said application on said smart card, said receiving placing said application into a loaded state;
installing said application on said smart card, said installing placing said application into an installed state;
registering said application on said smart card, said registering placing said application into a registered state;
personalizing said application on said smart card, said personalizing placing said application into a personalized state, whereby said application is available for use.

18. A method according to claim 17 further comprising the steps of:

receiving an application block request;
blocking said application; and,
placing said application into a blocked state, whereby said application is not available for use.

19. A method according to claim 17 or 18, further comprising the steps of:

receiving an application delete request;
deleting said application from said smart card; and,
indicating said application is in a not available state, whereby said application is not available for use.

20. A method according to any of claims 17 to 19, wherein said application is received by being loaded into a memory of said smart card during initialization of said smart card, whereby said application is present on said smart card before issuance.

21. A method according to any of claims 17 to 19, wherein said application is received by being loaded onto said smart card post-issuance, whereby said application appears on said smart card after issuance.

22. A method of moving an application of smart card through a sequence of application life cycle states after issuance of said smart card, said method comprising the steps of:

issuing said smart card;
indicating within said smart card that said application is in a not available state;
loading said application onto said smart card post-issuance, said loading placing said application into a loaded state; and,
installing said application on said smart card, said installing placing said application into an installed state, whereby said application is available for use on said smart card.

23. A method according to claim 22 further comprising the step of:
personalizing said application on said smart card, said personalizing placing said application into a personalized state.

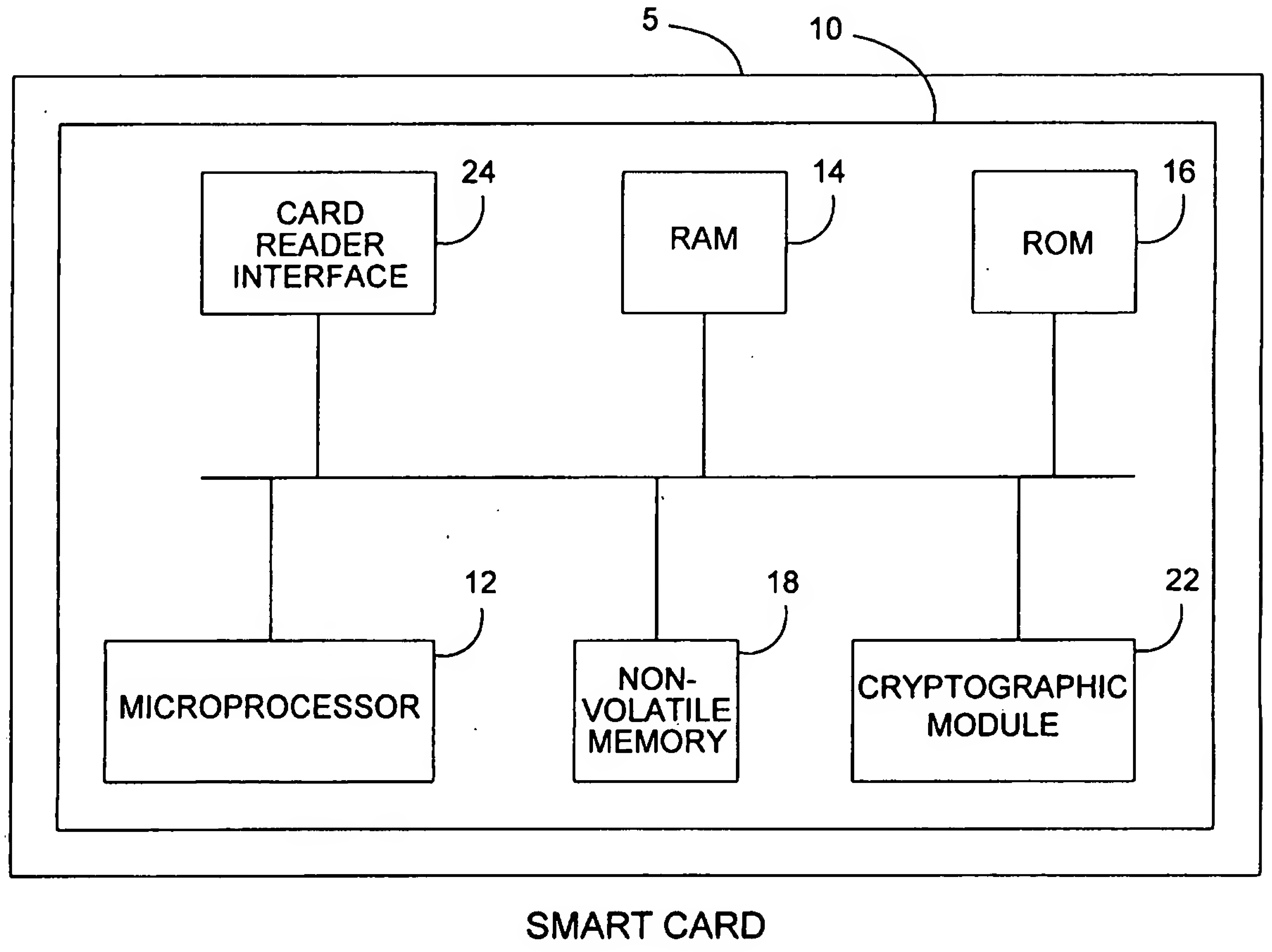
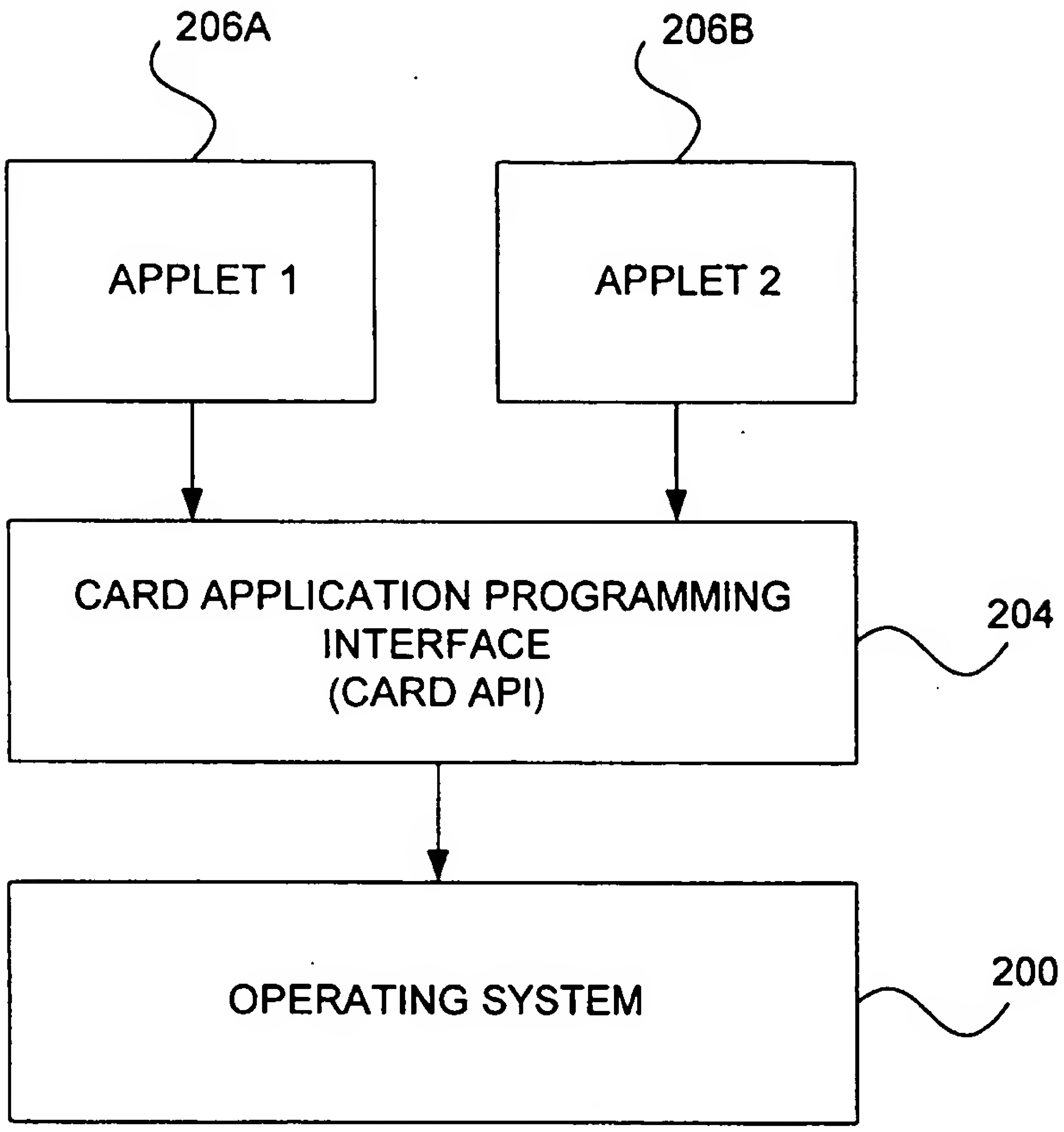


FIG. 1
(PRIOR ART)



SMART CARD SOFTWARE LAYERS

FIG. 2
(PRIOR ART)

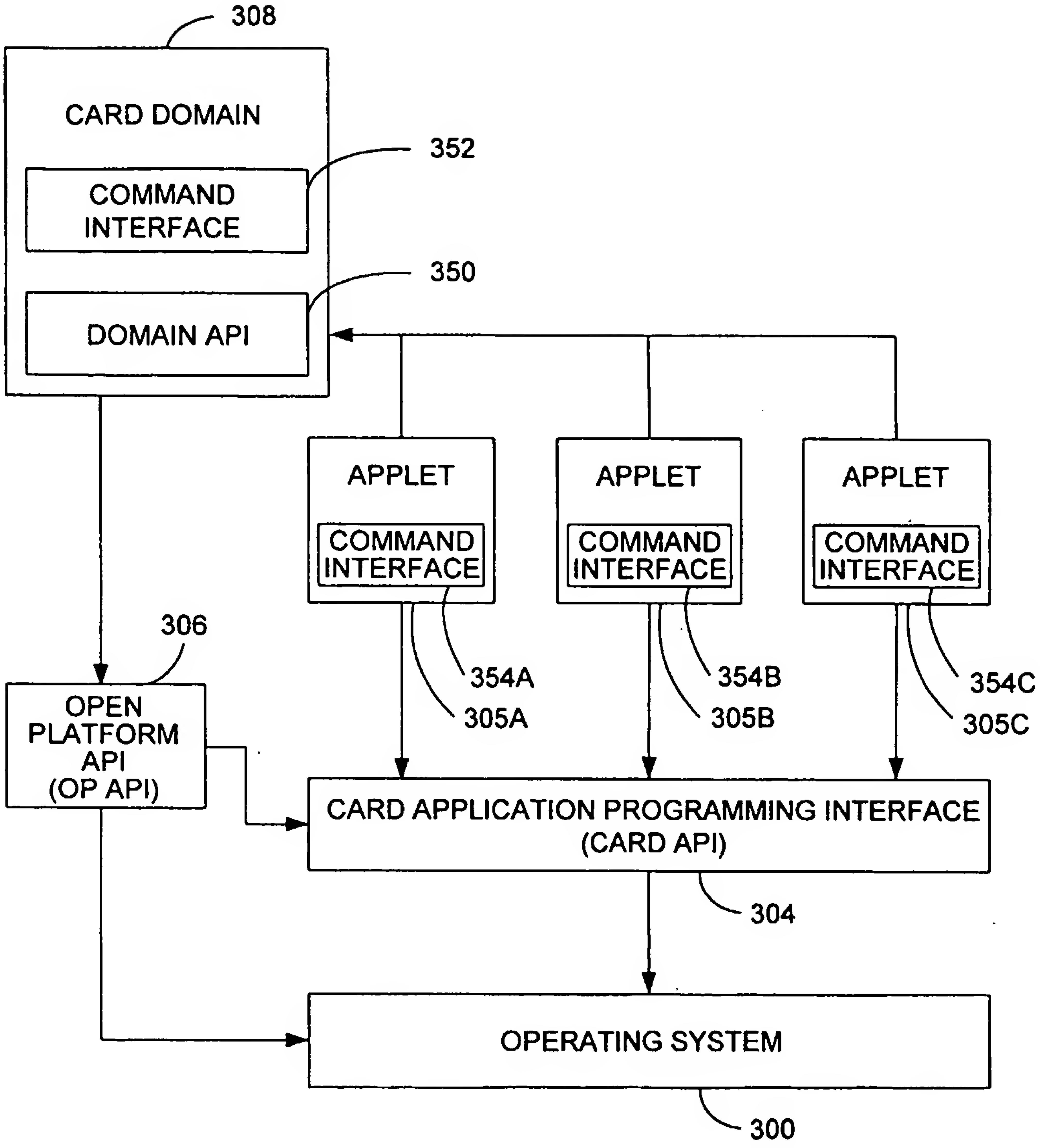

308
CARD DOMAIN
COMMAND INTERFACE
352
350
DOMAIN API
APPLET
COMMAND INTERFACE
APPLET
COMMAND INTERFACE
APPLET
COMMAND INTERFACE
306
354A
305A
354B
305B
354C
305C
OPEN PLATFORM API (OP API)
CARD APPLICATION PROGRAMMING INTERFACE (CARD API)
304
OPERATING SYSTEM
300

FIG. 3A

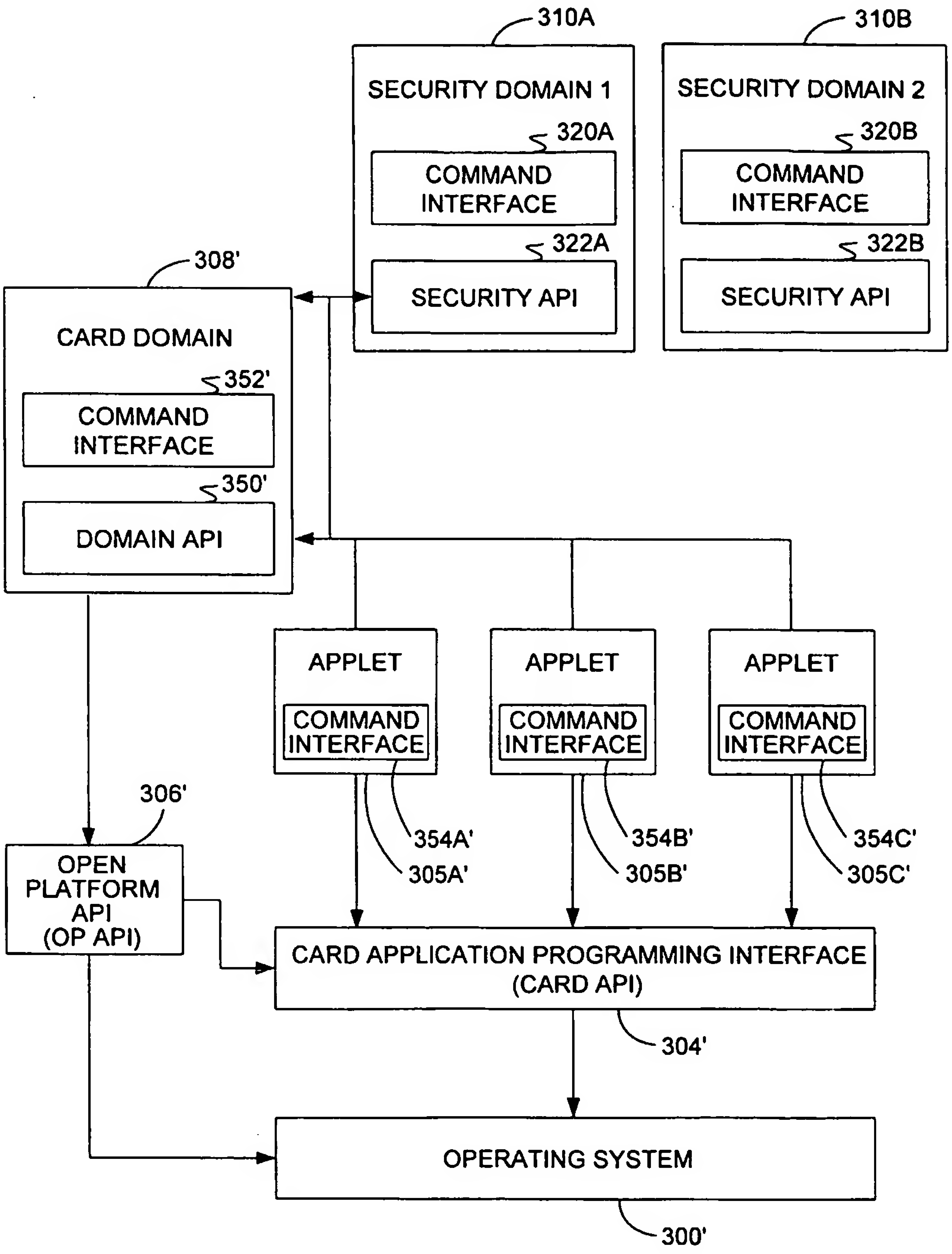

310A
310B
SECURITY DOMAIN 1
SECURITY DOMAIN 2
320A
320B
COMMAND INTERFACE
COMMAND INTERFACE
322A
322B
SECURITY API
SECURITY API
308'
CARD DOMAIN
352'
COMMAND INTERFACE
350'
DOMAIN API
APPLET
COMMAND INTERFACE
APPLET
COMMAND INTERFACE
APPLET
COMMAND INTERFACE
306'
OPEN PLATFORM API (OP API)
354A'
305A'
354B'
305B'
354C'
305C'
CARD APPLICATION PROGRAMMING INTERFACE (CARD API)
304'
OPERATING SYSTEM
300'

FIG. 3B

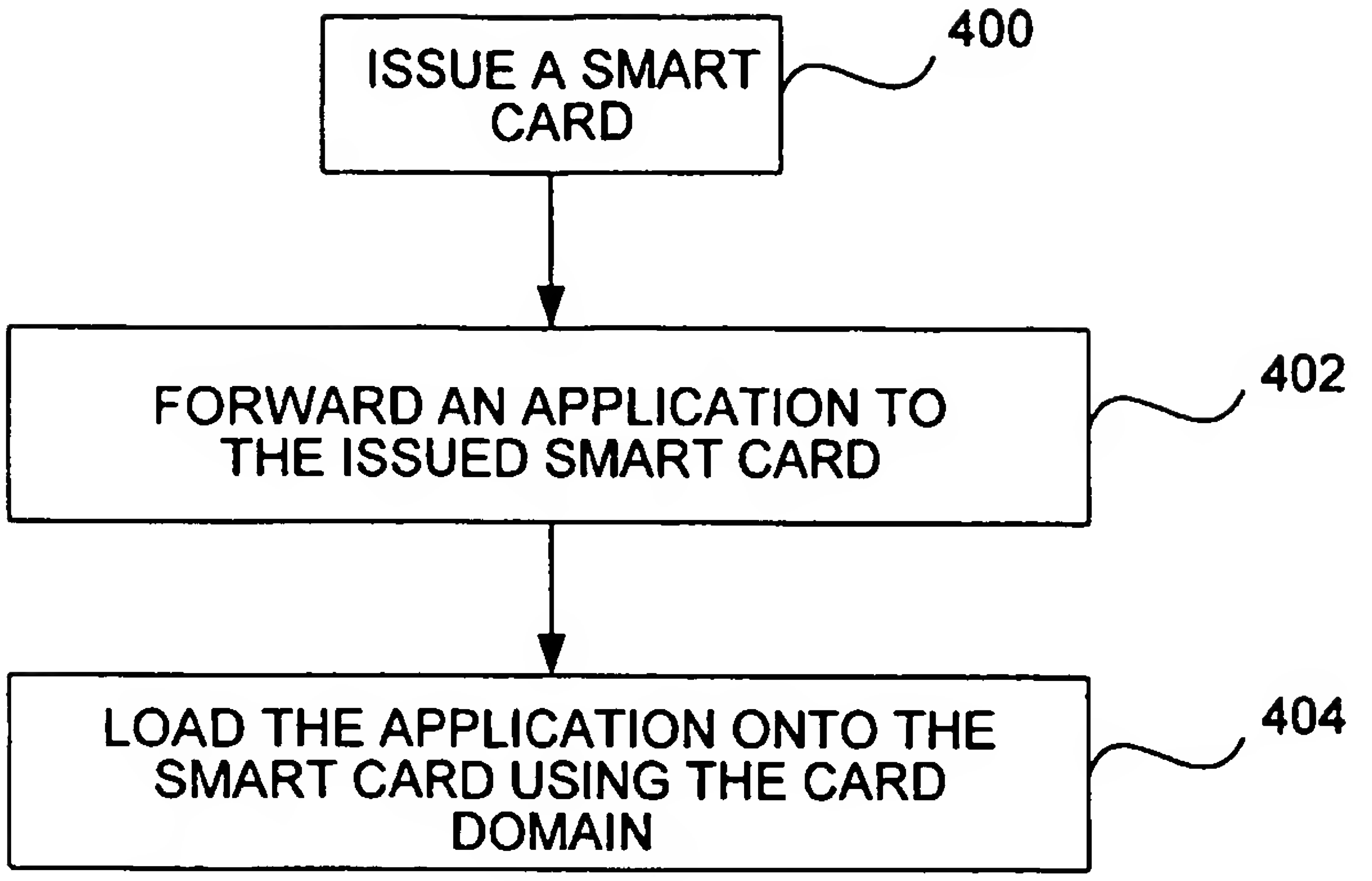

ISSUE A SMART CARD
400
FORWARD AN APPLICATION TO THE ISSUED SMART CARD
402
LOAD THE APPLICATION ONTO THE SMART CARD USING THE CARD DOMAIN
404

FIG. 4

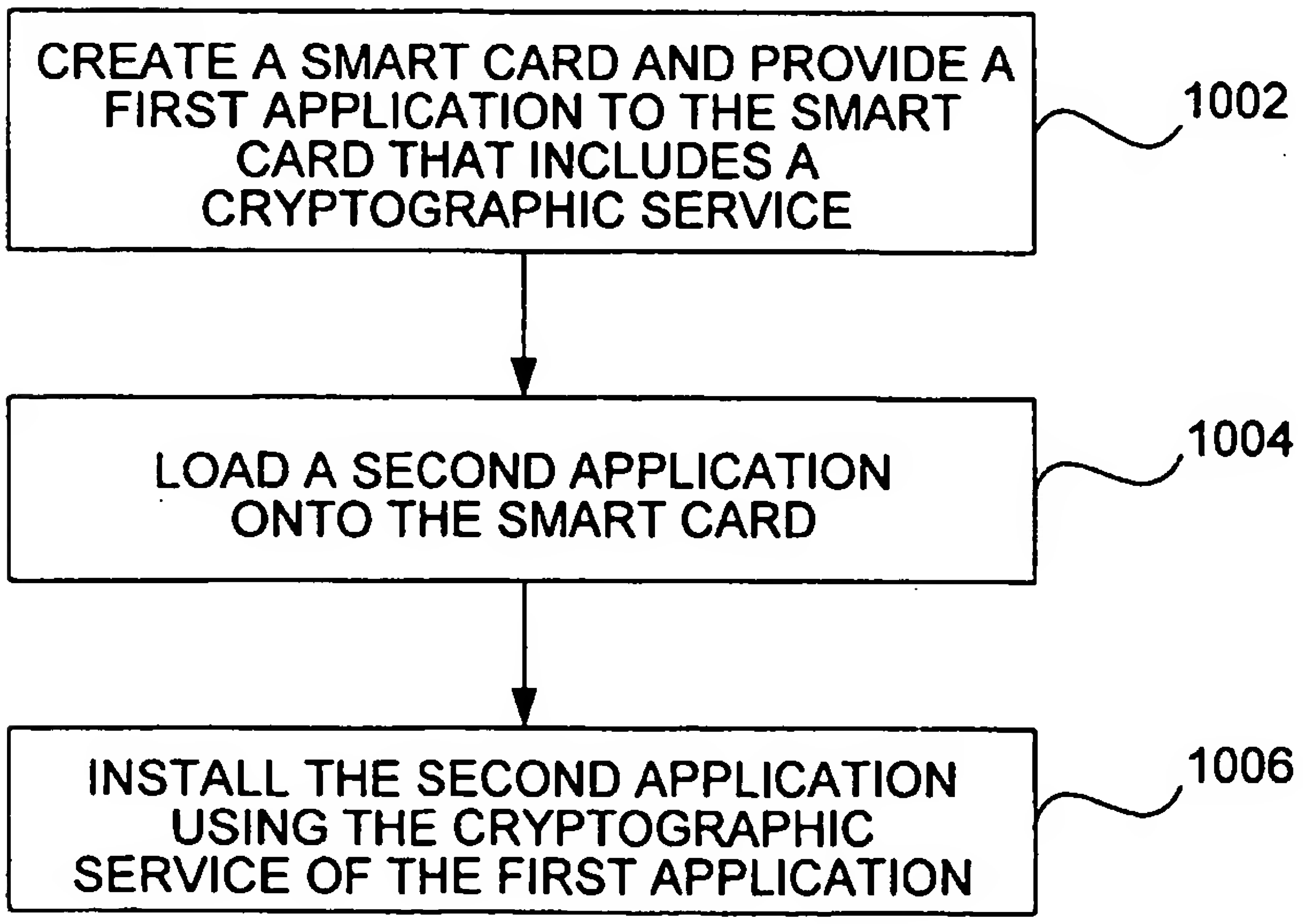

CREATE A SMART CARD AND PROVIDE A FIRST APPLICATION TO THE SMART CARD THAT INCLUDES A CRYPTOGRAPHIC SERVICE
1002
LOAD A SECOND APPLICATION ONTO THE SMART CARD
1004
INSTALL THE SECOND APPLICATION USING THE CRYPTOGRAPHIC SERVICE OF THE FIRST APPLICATION
1006

FIG. 5

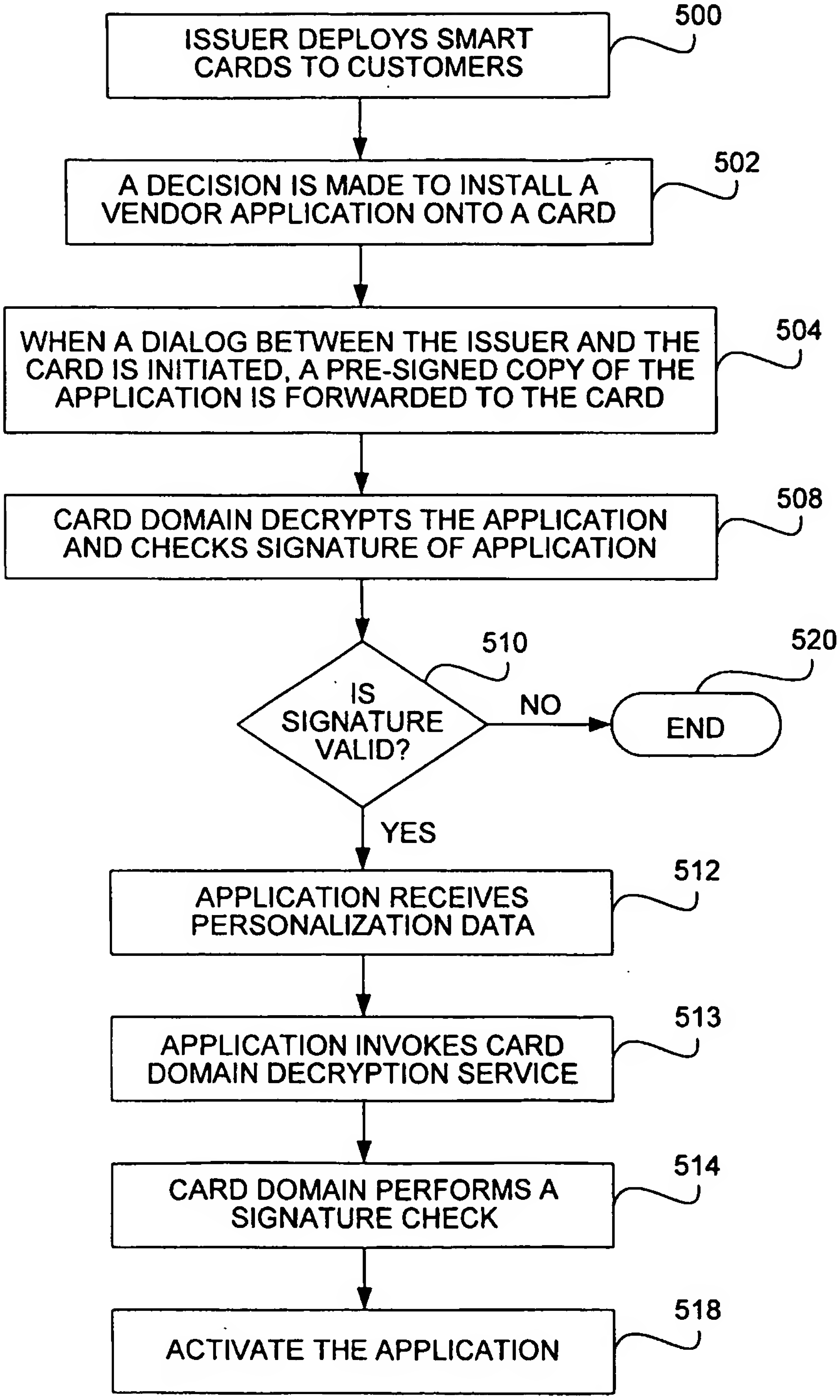

ISSUER DEPLOYS SMART CARDS TO CUSTOMERS
500
A DECISION IS MADE TO INSTALL A VENDOR APPLICATION ONTO A CARD
502
WHEN A DIALOG BETWEEN THE ISSUER AND THE CARD IS INITIATED, A PRE-SIGNED COPY OF THE APPLICATION IS FORWARDED TO THE CARD
504
CARD DOMAIN DECRYPTS THE APPLICATION AND CHECKS SIGNATURE OF APPLICATION
508
IS SIGNATURE VALID?
510
NO
END
520
YES
APPLICATION RECEIVES PERSONALIZATION DATA
512
APPLICATION INVOKES CARD DOMAIN DECRYPTION SERVICE
513
CARD DOMAIN PERFORMS A SIGNATURE CHECK
514
ACTIVATE THE APPLICATION
518

FIG. 6

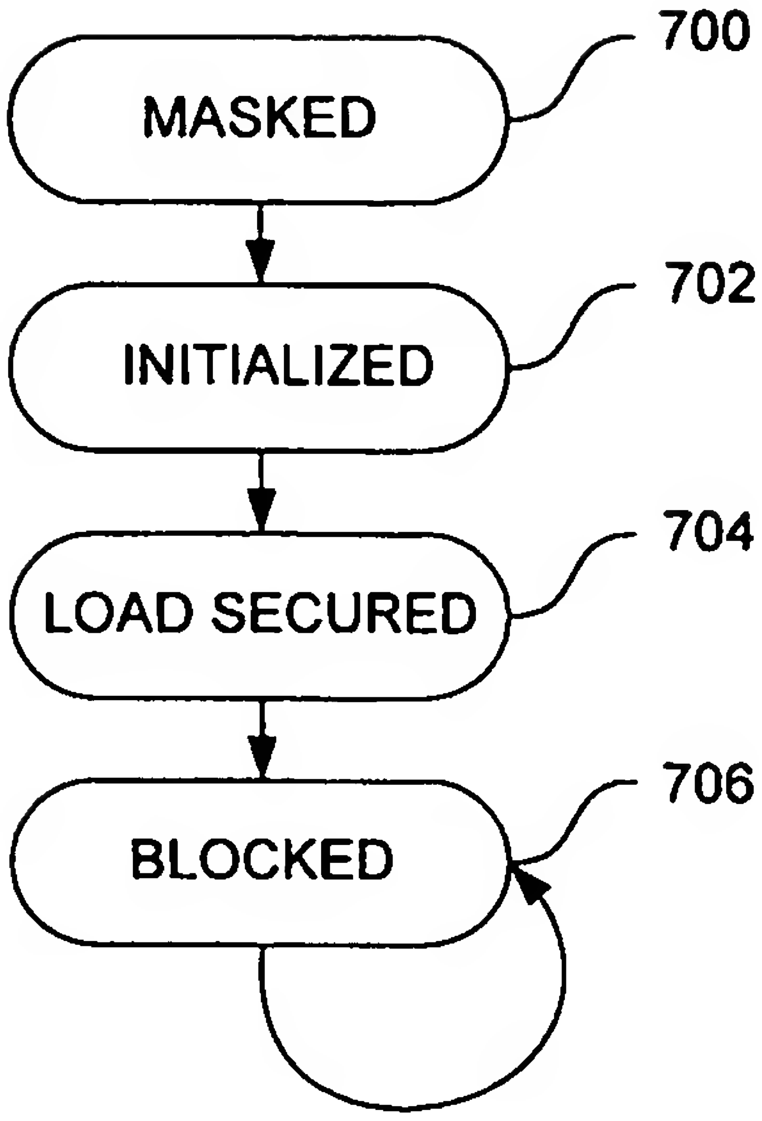

MASKED
700
INITIALIZED
702
LOAD SECURED
704
BLOCKED
706

FIG. 7A

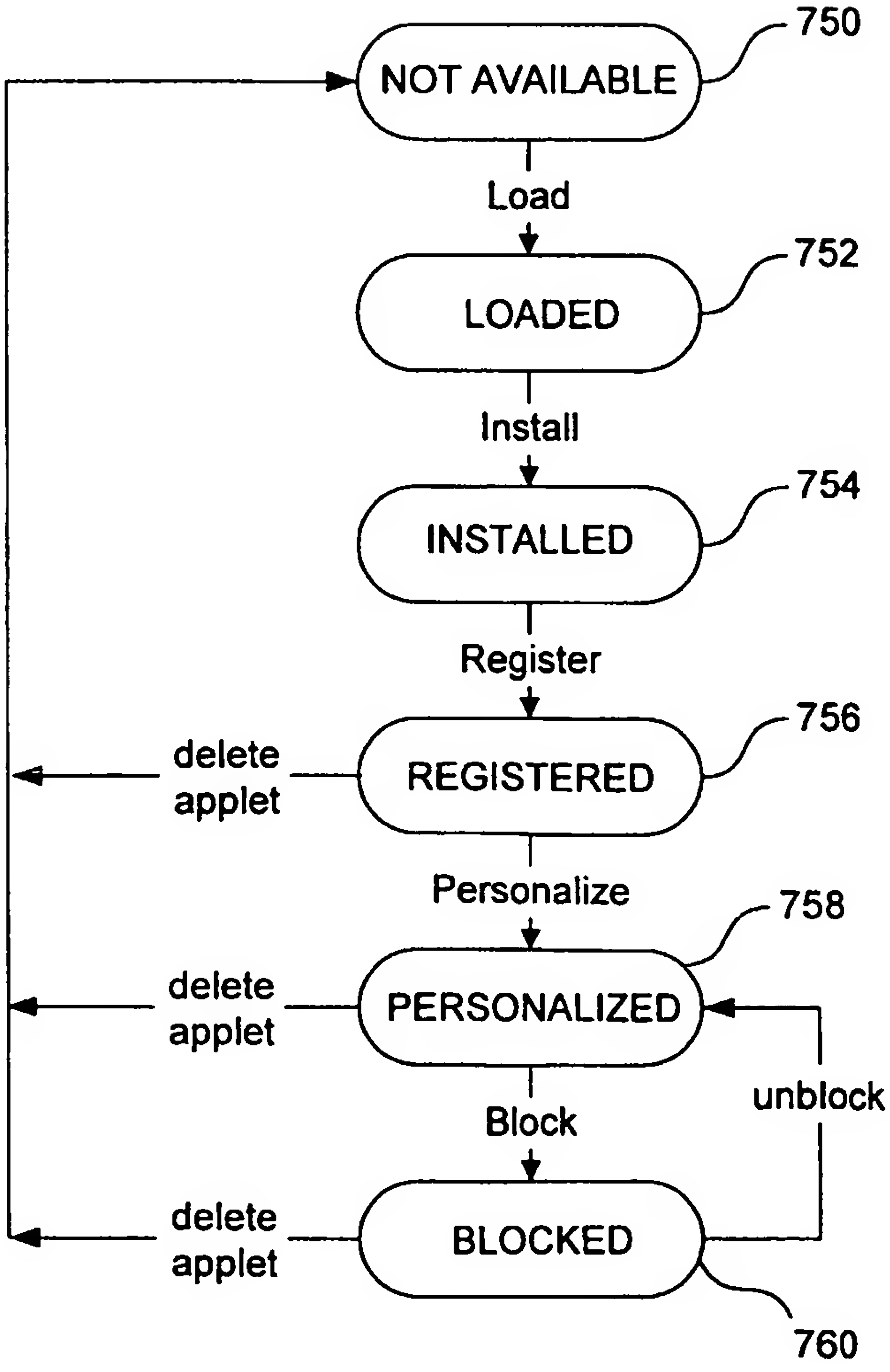

NOT AVAILABLE
750
Load
LOADED
752
Install
INSTALLED
754
Register
REGISTERED
756
delete applet
Personalize
PERSONALIZED
758
delete applet
Block
unblock
BLOCKED
760
delete applet

FIG. 7B

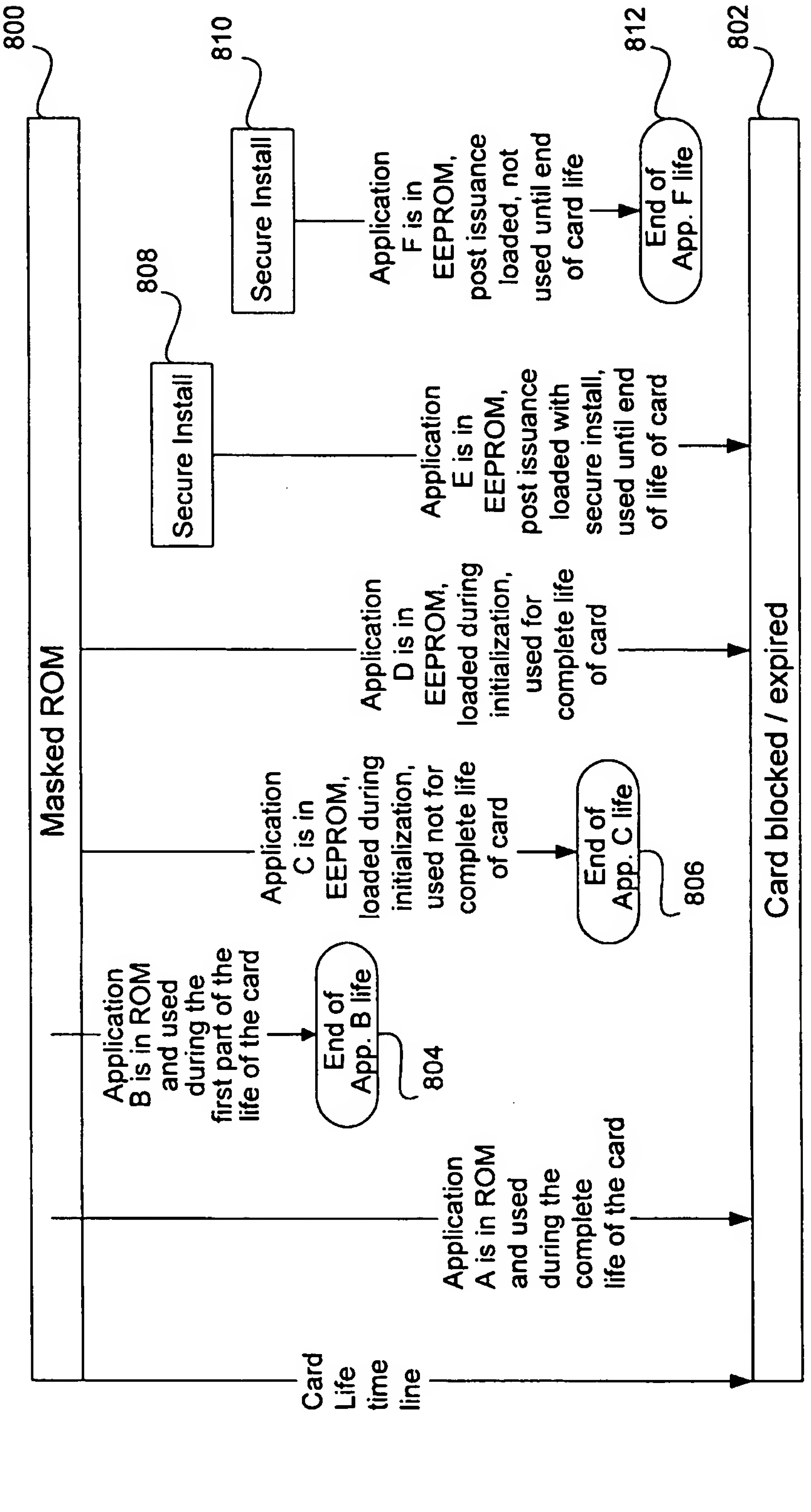

800
810
812
802
808
Masked ROM
Secure Install
Secure Install
Application F is in EEPROM, post issuance loaded, not used until end of card life
End of App. F life
Application E is in EEPROM, post issuance loaded with secure install, used until end of life of card
Application D is in EEPROM, loaded during initialization, used for complete life of card
Application C is in EEPROM, loaded during initialization, used not for complete life of card
End of App. C life
806
Application B is in ROM and used during the first part of the life of the card
End of App. B life
804
Application A is in ROM and used during the complete life of the card
Card Life time line
Card blocked / expired

FIG. 8

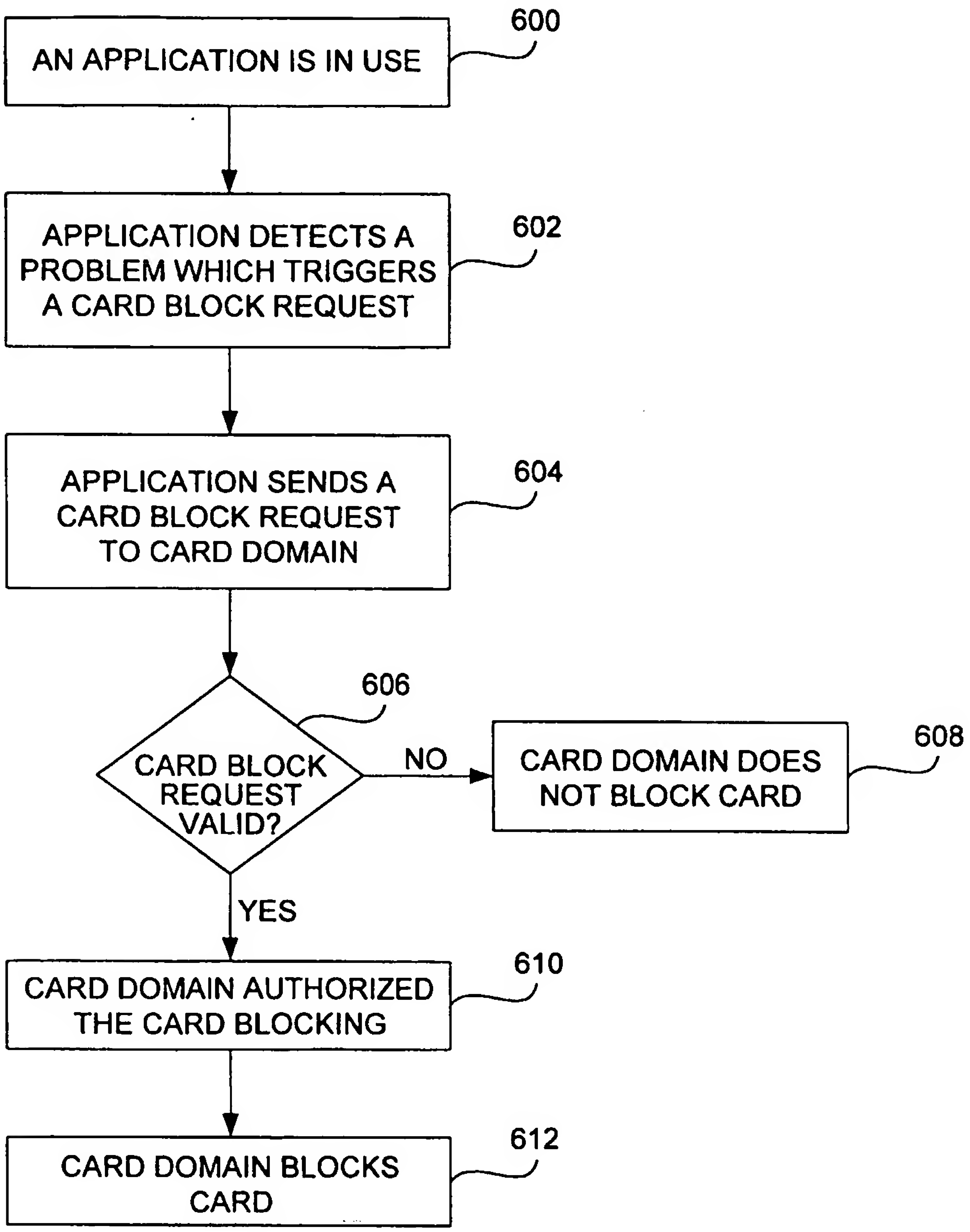

AN APPLICATION IS IN USE
600
APPLICATION DETECTS A PROBLEM WHICH TRIGGERS A CARD BLOCK REQUEST
602
APPLICATION SENDS A CARD BLOCK REQUEST TO CARD DOMAIN
604
CARD BLOCK REQUEST VALID?
606
NO
CARD DOMAIN DOES NOT BLOCK CARD
608
YES
CARD DOMAIN AUTHORIZED THE CARD BLOCKING
610
CARD DOMAIN BLOCKS CARD
612

FIG. 9

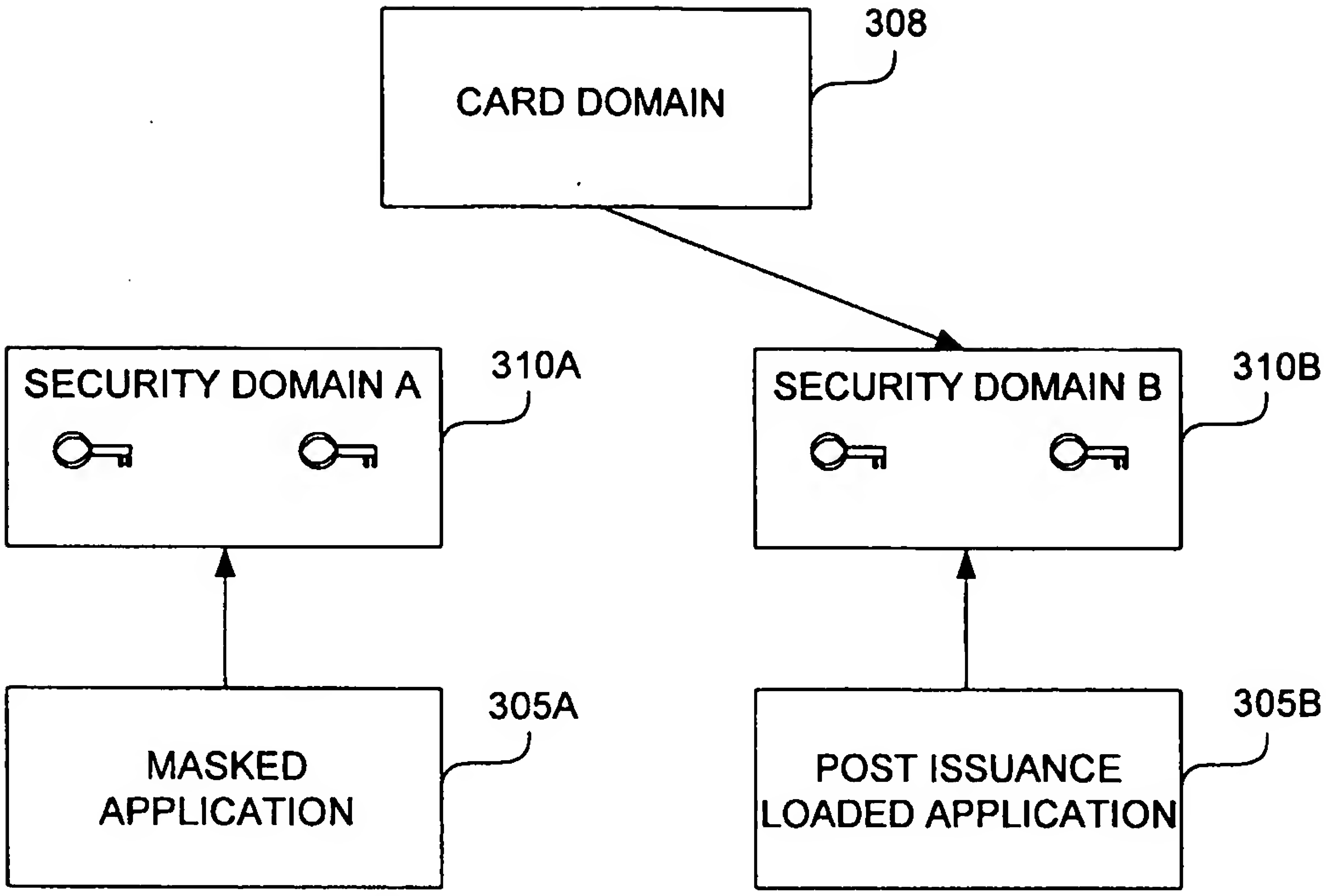

CARD DOMAIN
308
SECURITY DOMAIN A
310A
SECURITY DOMAIN B
310B
MASKED APPLICATION
305A
POST ISSUANCE LOADED APPLICATION
305B

FIG. 10

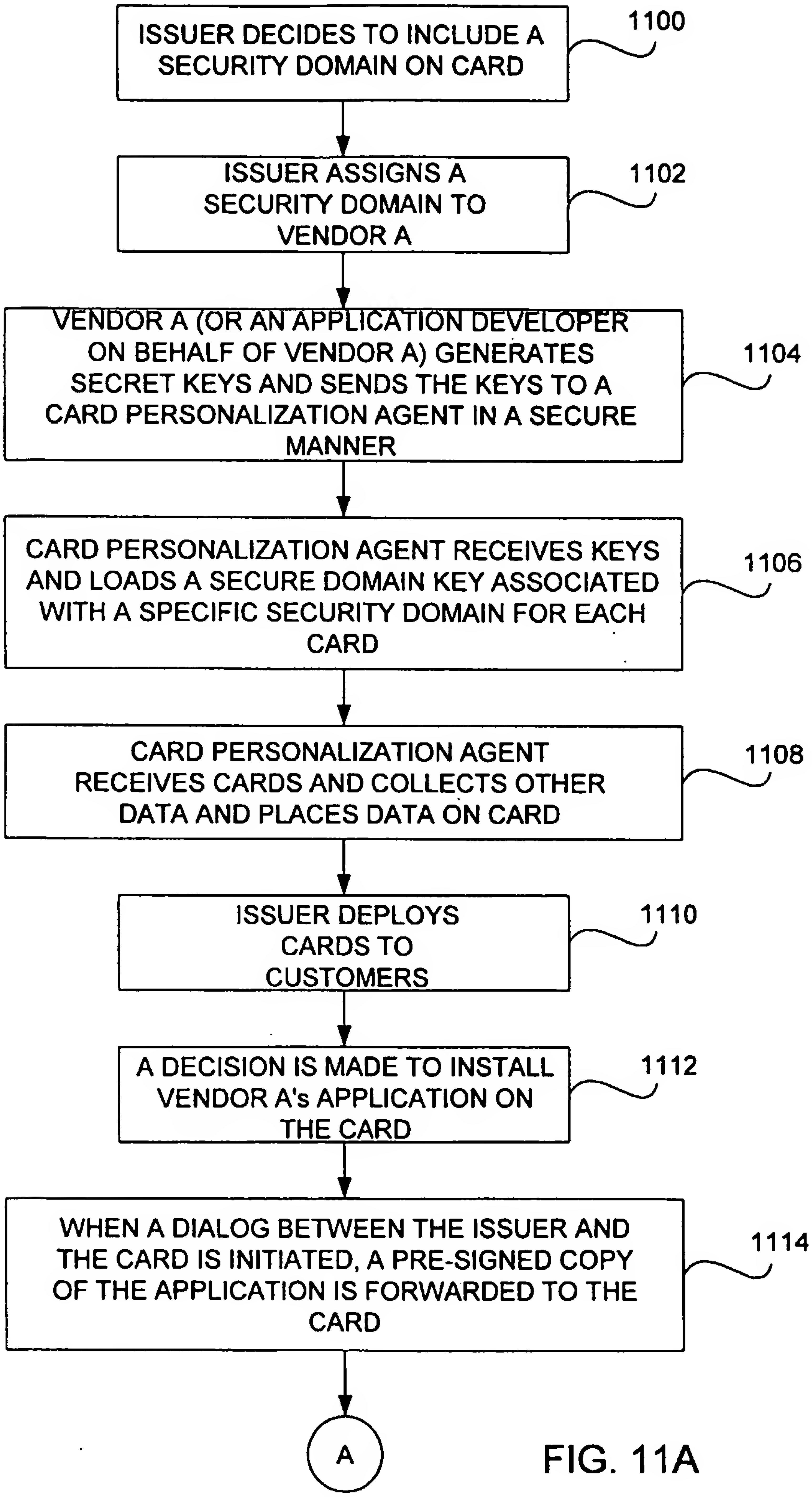

ISSUER DECIDES TO INCLUDE A SECURITY DOMAIN ON CARD
1100
ISSUER ASSIGNS A SECURITY DOMAIN TO VENDOR A
1102
VENDOR A (OR AN APPLICATION DEVELOPER ON BEHALF OF VENDOR A) GENERATES SECRET KEYS AND SENDS THE KEYS TO A CARD PERSONALIZATION AGENT IN A SECURE MANNER
1104
CARD PERSONALIZATION AGENT RECEIVES KEYS AND LOADS A SECURE DOMAIN KEY ASSOCIATED WITH A SPECIFIC SECURITY DOMAIN FOR EACH CARD
1106
CARD PERSONALIZATION AGENT RECEIVES CARDS AND COLLECTS OTHER DATA AND PLACES DATA ON CARD
1108
ISSUER DEPLOYS CARDS TO CUSTOMERS
1110
A DECISION IS MADE TO INSTALL VENDOR A's APPLICATION ON THE CARD
1112
WHEN A DIALOG BETWEEN THE ISSUER AND THE CARD IS INITIATED, A PRE-SIGNED COPY OF THE APPLICATION IS FORWARDED TO THE CARD
1114
A

FIG. 11A

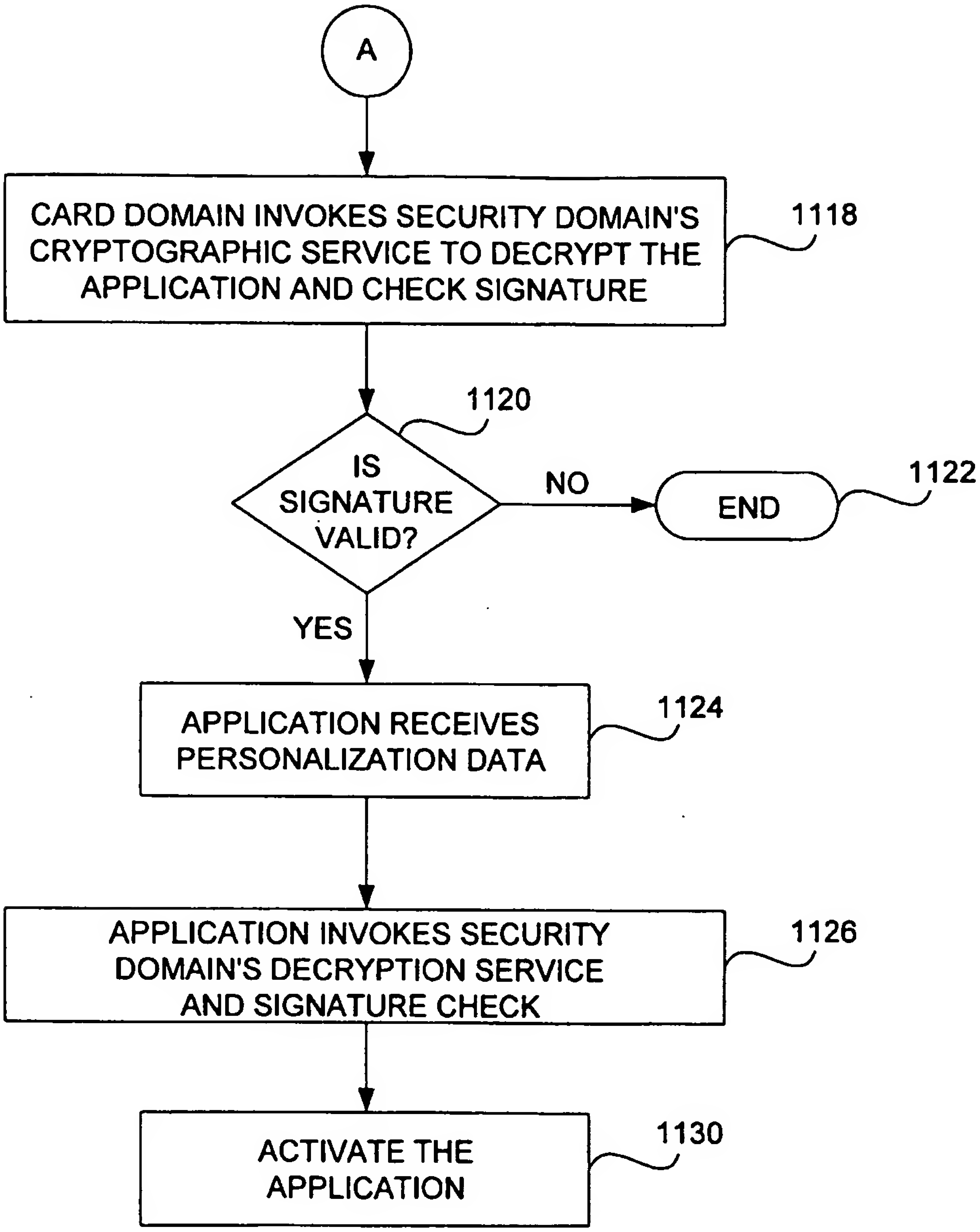

A
CARD DOMAIN INVOKES SECURITY DOMAIN'S CRYPTOGRAPHIC SERVICE TO DECRYPT THE APPLICATION AND CHECK SIGNATURE
1118
1120
IS SIGNATURE VALID?
NO
END
1122
YES
APPLICATION RECEIVES PERSONALIZATION DATA
1124
APPLICATION INVOKES SECURITY DOMAIN'S DECRYPTION SERVICE AND SIGNATURE CHECK
1126
ACTIVATE THE APPLICATION
1130

FIG. 11B

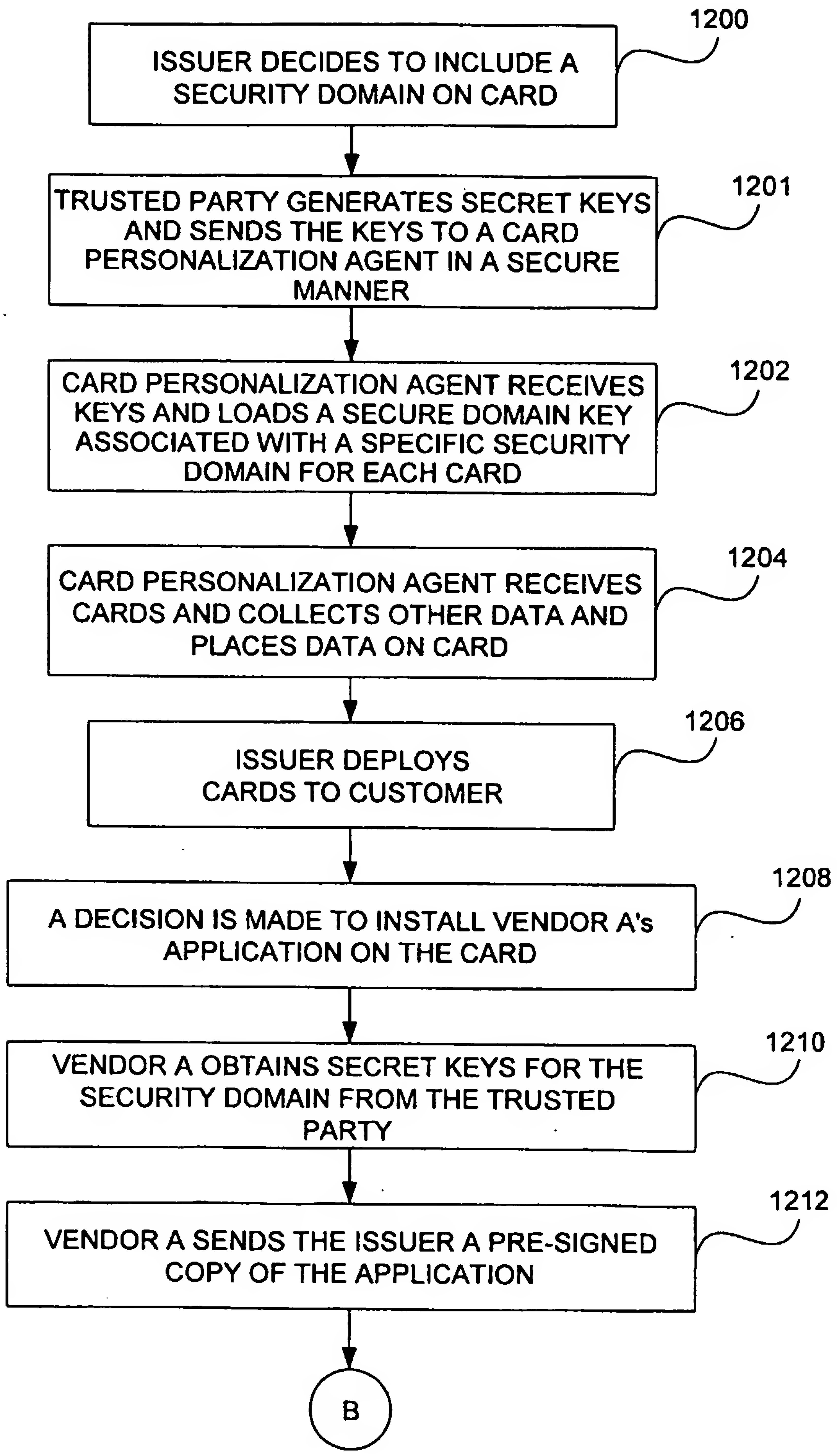

1200
ISSUER DECIDES TO INCLUDE A SECURITY DOMAIN ON CARD
1201
TRUSTED PARTY GENERATES SECRET KEYS AND SENDS THE KEYS TO A CARD PERSONALIZATION AGENT IN A SECURE MANNER
1202
CARD PERSONALIZATION AGENT RECEIVES KEYS AND LOADS A SECURE DOMAIN KEY ASSOCIATED WITH A SPECIFIC SECURITY DOMAIN FOR EACH CARD
1204
CARD PERSONALIZATION AGENT RECEIVES CARDS AND COLLECTS OTHER DATA AND PLACES DATA ON CARD
1206
ISSUER DEPLOYS CARDS TO CUSTOMER
1208
A DECISION IS MADE TO INSTALL VENDOR A's APPLICATION ON THE CARD
1210
VENDOR A OBTAINS SECRET KEYS FOR THE SECURITY DOMAIN FROM THE TRUSTED PARTY
1212
VENDOR A SENDS THE ISSUER A PRE-SIGNED COPY OF THE APPLICATION
B

FIG. 12A

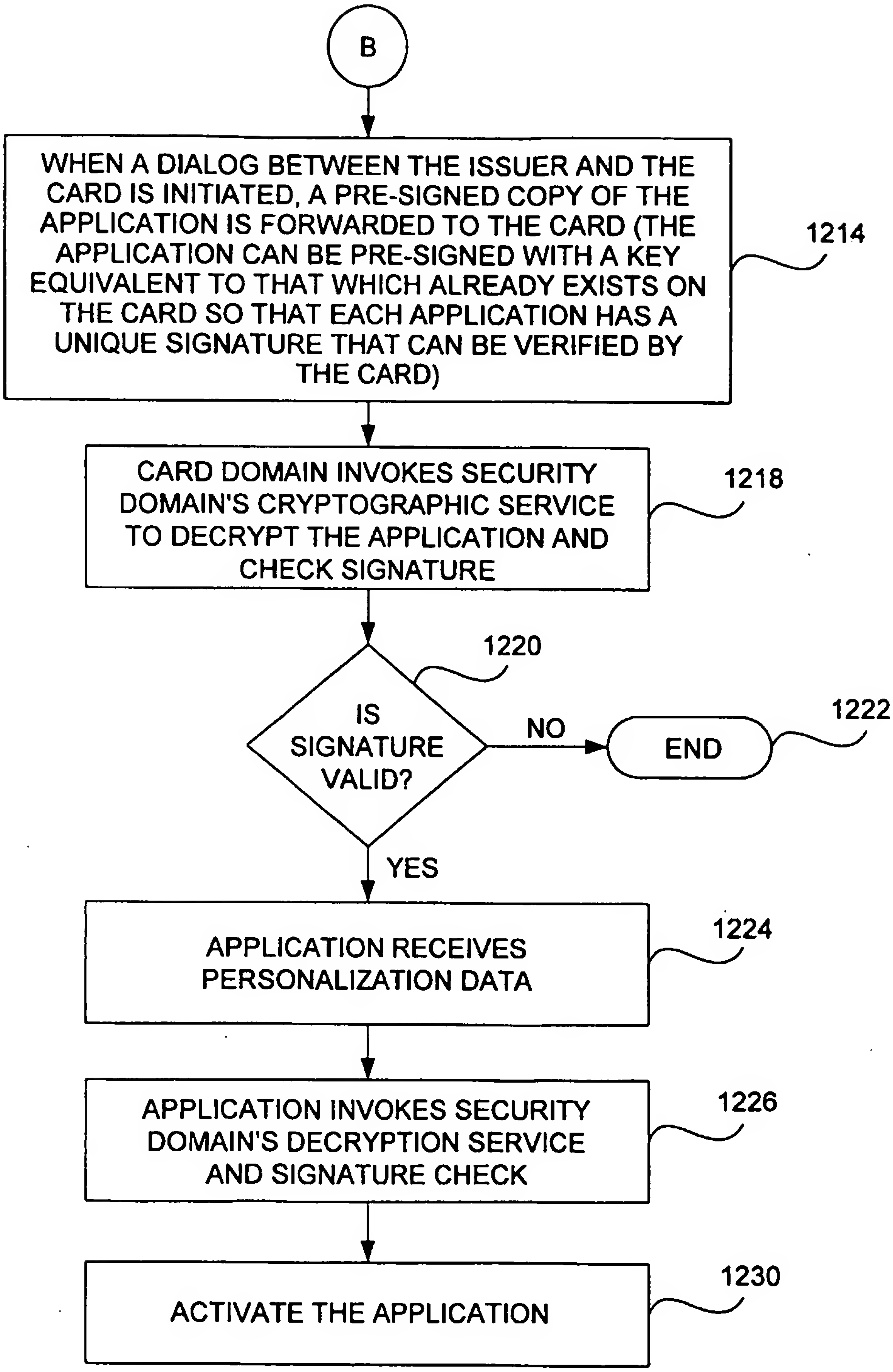

B
WHEN A DIALOG BETWEEN THE ISSUER AND THE CARD IS INITIATED, A PRE-SIGNED COPY OF THE APPLICATION IS FORWARDED TO THE CARD (THE APPLICATION CAN BE PRE-SIGNED WITH A KEY EQUIVALENT TO THAT WHICH ALREADY EXISTS ON THE CARD SO THAT EACH APPLICATION HAS A UNIQUE SIGNATURE THAT CAN BE VERIFIED BY THE CARD)
1214
CARD DOMAIN INVOKES SECURITY DOMAIN'S CRYPTOGRAPHIC SERVICE TO DECRYPT THE APPLICATION AND CHECK SIGNATURE
1218
1220
IS SIGNATURE VALID?
NO
END
1222
YES
APPLICATION RECEIVES PERSONALIZATION DATA
1224
APPLICATION INVOKES SECURITY DOMAIN'S DECRYPTION SERVICE AND SIGNATURE CHECK
1226
ACTIVATE THE APPLICATION
1230

FIG. 12B

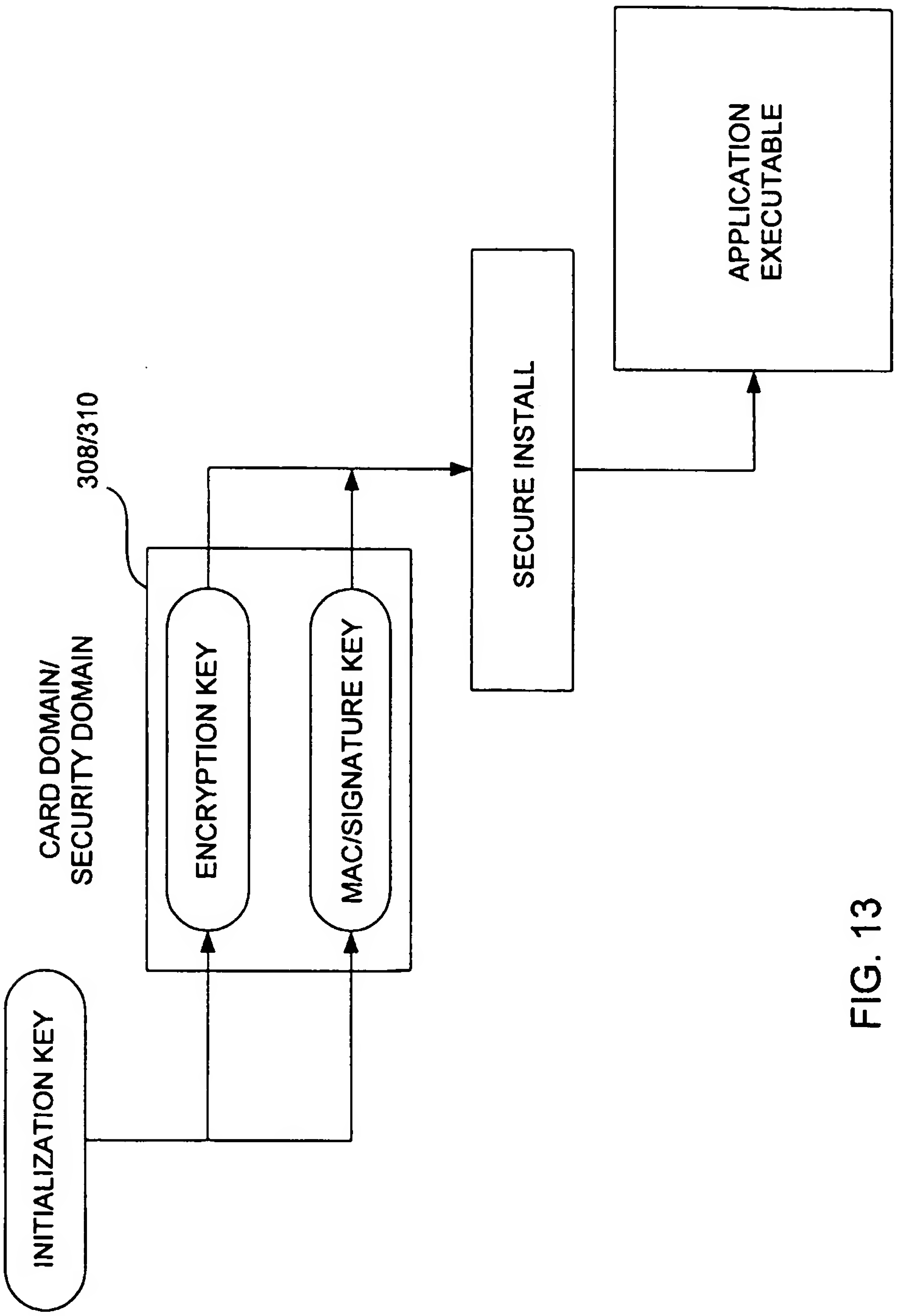

INITIALIZATION KEY
CARD DOMAIN/
SECURITY DOMAIN
308/310
ENCRYPTION KEY
MAC/SIGNATURE KEY
SECURE INSTALL
APPLICATION
EXECUTABLE

FIG. 13